



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKTIVNÍ HRA S PRVKY DOBRODRUŽSTVÍ

A DETECTIVE GAME WITH ELEMENTS OF ADVENTURE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DINARA GARİPOVA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ MILET, Ph.D.

BRNO 2024

Zadání bakalářské práce



153525

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Studentka: **Garipova Dinara**
Program: Informační technologie
Název: **Detektivní hra s prvky dobrodružství**
Kategorie: Počítačová grafika
Akademický rok: 2023/24

Zadání:

1. Nastudujte herní vývoj, herní enginy a existující hry s logickými hádankami.
2. Navrhněte dobrodružnou hru založenou na logických hádankách. Navrhněte vhodné herní mechaniky logických her.
3. Vytvořte nebo najděte vhodné herní objekty. Implementujte logické hádanky a celkovou hru podle návrhu. Průběžně hru testujte na uživatelích.
4. Hru otestujte a proměřte.
5. Sepište závěry, možnosti rozšíření, hru zveřejněte a vytvořte demonstrační video.

Literatura:

- Ernest Adams. Fundamentals of Game Design 3rd Edition. New Riders 2013. ISBN-100321929675.

Při obhajobě semestrální části projektu je požadováno:

Funkční prototyp a čtyři logické hádanky. Třicet stran technické dokumentace.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Milet Tomáš, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 9.5.2024
Datum schválení: 9.11.2023

Abstrakt

Cílem práce je navrhnout a vytvořit detektivní počítačovou hru s prvky dobrodružného žánru, která má hráče hluboce vtáhnout do interaktivního příběhu s využitím mechanik vizuálního románu a hádanek. Hra se skládá z hlavní a vedlejší části, přičemž v té první je hráč prostřednictvím dialogů vtažen do příběhu. Hlavním úkolem hráče je vyřešit záhadu starobylé vesnice pomocí indicií a řešení hádanek. Vedlejší část tvoří čtyři různé minihry (Skládačka, Otevřené bludiště, Najdi dvojici (Pexeso) a Běžec) zaměřené na trénování paměti, logického myšlení a rychlosti reakcí. Výsledná aplikace byla vytvořena pomocí herního vývojového enginu Unity a testována pomocí formulářů Google.

Abstract

The aim of the work is to design and create a detective computer game with elements of the adventure genre, designed to deeply immerse players in an interactive story using visual novel mechanics and puzzles. The game consists of a main and a secondary part, where in the former the player is immersed in the story through dialogues. The main task of the player is to solve the mystery of the ancient village using clues and solving puzzles. The secondary parts consist of various mini-games aimed at training memory, logical thinking and reaction speed. The resulting application was created using the Unity game development engine and tested with Google forms.

Klíčová slova

Detektivní hra, dobrodružný žánr, interaktivní příběh, vizuální román, hádanky, minihry, trénování paměti, logické myšlení, 2D, herní engine Unity, vývoj počítačových her

Keywords

Detective game, adventure genre, interactive story, visual novel, puzzles, mini-games, memory training, logical thinking, 2D, Unity game engine, computer game development.

Citace

GARIPOVA, Dinara. *Detektivní hra s prvky dobrodružství*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Milet, Ph.D.

Detektivní hra s prvky dobrodružství

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doktora Tomáše Mileta. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....
Dinara Garipova
8. května 2024

Poděkování

Chtěla bych upřímně poděkovat všem, kteří se podíleli na testování mé hry. Váš drahocenný čas a pozornost věnovaná detailům byly při dokončování projektu neocenitelné. Zvláštní poděkování patří mé rodině a přátelům, jejichž neochvějná podpora a víra ve mne mi sloužily jako opora po celou dobu práce na diplomu. Avšak největší díky patří vedoucímu mé bakalářské práce, Tomášu Miletovi. Jeho neocenitelné rady, ochota diskutovat o obtížných otázkách a pozitivní přístup byly základem mého výzkumu. Chci také vyjádřit svou hlubokou vděčnost své přítelkyni Urazbaevové Anastasii, jejíž talent a dovednost při tvorbě pozadí pro hru se ukázaly být neocenitelným přínosem pro vizuální složku projektu. Její úsilí výrazně obohatilo atmosféru hry a přispělo k její výraznosti.

Obsah

1	Úvod	2
2	Detektivní hry, jejich vývoj a typy	4
2.1	Vývoj počítačových her	4
2.2	Hádanky a jejich vývoj	6
2.3	Druhy detektivních her	8
2.4	Analýza podobných her	9
3	Herní enginey a Unity pro vytváření detektivních her	14
3.1	Herní enginey	14
3.2	Entity-Component-System (ECS) v herních enginech:	15
3.3	Výhody Unity pro vytváření detektivních her	16
3.4	Unity Engine	17
3.5	Životní cyklus projektu v Unity	25
4	Návrh, koncept a logická struktura hry	28
4.1	Průběh hry	29
4.2	Úvodní menu	30
4.3	Nastavení	30
4.4	Hlavní hra	31
4.5	Minihry	33
5	Realizace	40
5.1	Hlavní hra	41
5.2	Minihry	45
5.3	Úvodní menu a nastavení	55
6	Uživatelské testování	57
6.1	Metodika	57
6.2	Výsledky	57
7	Závěr	63
	Literatura	64

Kapitola 1

Úvod

„Hra je sled zajímavých voleb.“
Sid Meier [11]

Videohra nebo počítačová hra je elektronická hra, která zahrnuje interakci s uživatelským rozhraním nebo vstupním zařízením (jako je joystick, ovladač, klávesnice nebo zařízení pro detekci pohybu) za účelem vytvoření vizuální zpětné vazby ze zobrazovacího zařízení, nejčastěji zobrazeného ve formátu videa na televizoru, monitoru počítače, ploché obrazovce nebo dotykovém displeji přenosných zařízení nebo náhlavní soupravě pro virtuální realitu¹.

Moderní herní průmysl je velmi rozsáhlý a poskytuje hráčům rozmanité virtuální světy. V tomto kontextu zaujímají zvláštní místo detektivní hry, které nabízejí kombinaci interaktivity a intelektuální výzvy.

„*Jaká je dobrá detektivní hra? Záhada, kterou je třeba vyřešit, soubor náročných, ale přesto zábavných hádanek a úkolů, které je třeba splnit.*“ [1]

To znamená, že pojem detektivní hry zahrnuje širokou škálu žánrů, v nichž je klíčový důraz kladen na vyšetřování a řešení záhad. Tyto hry vnášejí do herního zážitku prvky intrik, logického myšlení a kreativity.

Jednou z výhod detektivních her je jejich schopnost vtáhnout hráče do zápletky, která vyžaduje nejen rychlé reakce, ale také hlubokou analýzu událostí. Na rozdíl od mnoha jiných žánrů detektivní hry často vyzývají hráče, aby řešil složité hádanky, řešil záhady a sledoval detaily zápletky. Tato vlastnost dává hře jedinečnou výhodu a činí ji atraktivní pro ty, kteří hledají kombinaci intelektuální výzvy a zábavy.

Cílem této práce je vytvořit detektivní hru s prvky dobrodružství. Události se odehrají v naší době. Dospívající dívka Olivia je kvůli svému psychickému stavu poslána na celé léto k prarodičům do starobylé, téměř opuštěné vesnice, o které kolují různé podivné pověsti.

Aniž by to čekala, stane se hlavní postavou skutečné detektivky, ruku v ruce se svým novým známým – kouzelným gnómem strážným – stojí před nelehkým úkolem, projít mnoha zkouškami, vyřešit hádanky a zachránit malou vesnici.

Samotná hra se skládá ze dvou hlavních částí: hlavního děje v podobě vizuálního románu a hádanek v podobě miniher zakomponovaných do děje. Součástí hlavního příběhu je vizuální román s interaktivními možnostmi, které vedou příběh na základě rozhodnutí hráče. To dává hráči možnost nejen si příběh užít, ale také ho sám ovlivnit, čímž se každé

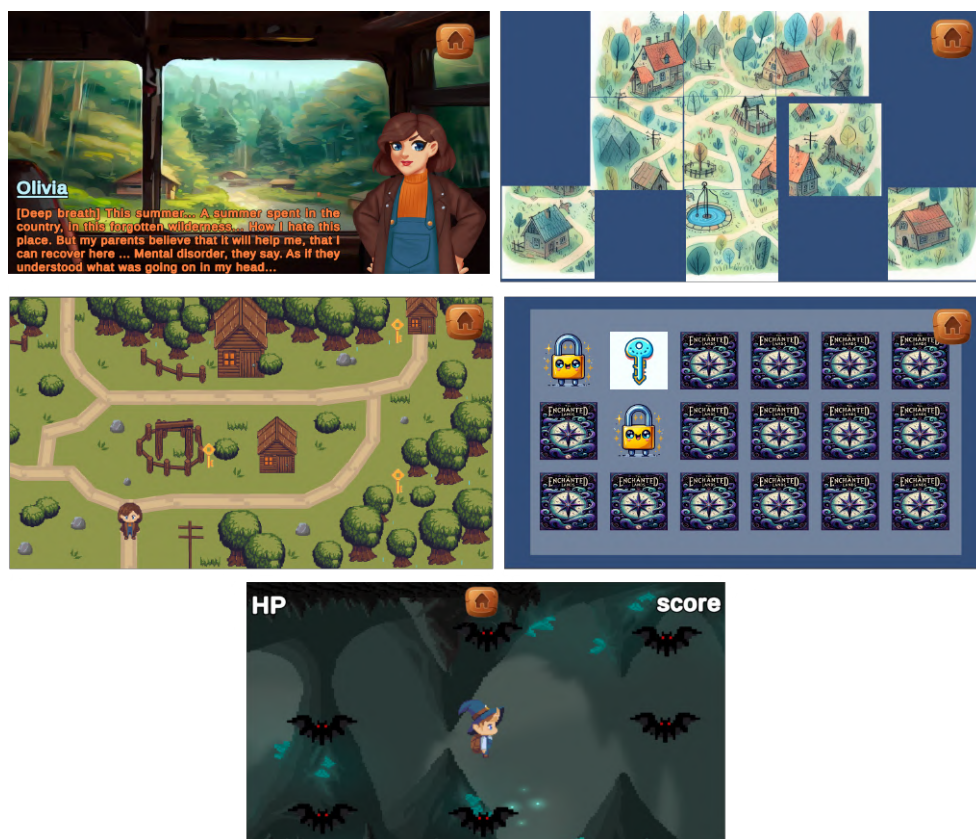
¹Odkaz na definice z Wikipedie – https://en.wikipedia.org/wiki/Video_game

hraní hry stává výjimečným. Vedlejší části zase představují intelektuální výzvy v podobě hádanek, logických problémů nebo hledání klíčových vodítek. Jejich integrace do celkového scénáře nejen zpestří hru, ale bude také vyžadovat originální myšlení hráče a schopnost rychle řešit hádanky.

Implementace hlavní herní funkcionality byla založena na technice *pixel art*². To znamená, že se hra při řešení hádanek nesoustředí na postavy, aby hráč nebyl rozptylován detaily a zůstal soustředěný. Nicméně prvky *digital art*³ byly použity k maximalizaci ponoření hráče do vzrušujícího děje. Tento přístup umožňuje hráči získat správnou představu o herním světě a ponořit se do příběhu, jako by se cítil na místě hlavní postavy.

Pro implementaci pomocných částí hry bylo nutné integrovat funkcionalitu řešení hádanek, procházení bludištěm, hledání stejných obrázků a uhýbání překážkám.

Výslednou aplikaci si můžete prohlédnout na obrázku 1.1. Kapitola 2 podrobně prozkoumá fáze tvorby počítačové hry a zejména detektivní hry a hlavolamy, které tvoří převážnou část všech detektivek. Také analyzovány typy detektivních her a hry, které inspirovaly vznik *The Secret of Summerside: Olivia's Mysterious Journey*. Kapitola 3 se zabývá populárními herními enginey a zejména enginem Unity. Rozebírá jeho součásti, architekturu a výhody pro tvorbu detektivních her. Návrh je detailně popsán v kapitole 4 Kapitola 5 se zaměřuje na implementaci pomocí herního vývojového enginu Unity a analyzuje, jak je zajištěna grafická stránka hry. Poslední kapitola, tj. kapitola 6, se zabývá testováním.



Obrázek 1.1: Ukázka hry : Hlavní hra s dialogy, minihra s hádankami, minihra s bludištěm, minihra najdi dvojici a minihra běžec.

²Odkaz na definice *pixel artu* z Wikipedie – https://cs.wikipedia.org/wiki/Pixel_art

³Odkaz na definice *digital artu* z Wikipedie – https://en.wikipedia.org/wiki/Digital_art

Kapitola 2

Detektivní hry, jejich vývoj a typy

Úspěch každé z nejslavnějších her začal jejich vývojem. Proces tvorby počítačových her je složitá cesta zahrnující kreativní návrh, technické provedení a strategické plánování. Každá fáze vývoje hry, od nápadu a počátečních skic až po závěrečné testování a vydání, s sebou nese vlastní výzvy a příležitosti. V tomto ohledu existují hlavní milníky, které jsou klíčové pro úspěšné vydání hry. Právě v těchto fázích se vytvářejí základy budoucího úspěchu, který světu poskytne nové, kvalitní herní zážitky.

Tato kapitola podrobně prozkoumá fáze tvorby počítačové hry a zejména detektivní hry a hlavolamy, které tvoří převážnou část všech detektivek. Analyzovány také typy detektivních her a hry, které inspirovaly vznik *The Secret of Summerside: Olivia's Mysterious Journey*.

2.1 Vývoj počítačových her

Vývoj her je proces tvorby her, popisující design, vývoj a vydání hry. Při vytváření hry je důležité myslet na herní mechanismy, odměny, zapojení hráčů a design úrovní [5]. V této části podrobně popsány kroky při vytváření hry (viz Obrázek 2.1):

- **Plánování**
- **Předprodukce**
- **Produkce**
- **Testování**
- Před vydáním
- **Vydání**
- Po vydání



Obrázek 2.1: Fáze vývoje hry. Na obrázku je vizualizována struktura fází vývoje hry, která zahrnuje klíčové kroky od počátečního plánování až po konečné fázi podpory po vydání.

2.1.1 Plánování

Před zahájením vývoje hry je třeba zodpovědět několik základních otázek [3]:

1. V jakém žánru bude hra?
2. Bude to 2D nebo 3D?
3. Jaký umělecký styl bude použit?
4. Jaké herní mechanismy budou zahrnuty?
5. Kdo je hrdina a kdo padouch?
6. Jaký herní engine bude použit?

Toto je jedna z nejdůležitějších částí procesu vývoje videoher. Provádění změn v základech může později vést k vážným problémům.

2.1.2 Předprodukce

Druhá fáze vývoje hry, známá jako předprodukční fáze, je místem, kde začíná skutečná činnost. Stanoví se rozsah práce, vytvoří se scénáře, prototypy a pak se rozhoduje o tom, které nápady budou úspěšné a které se bohužel upraví.

Pokud hru vytváří pouze jeden vývojář, musí se ujistit, že vzal v úvahu všechny potřebné detaily, než se pustí do samotné tvorby hry.

Je třeba se ujistit, že výtvarný styl a barevná paleta odpovídají tématu a žánru hry. Možná by stálo za to zvážit vytvoření vlastního stylu. Pak je třeba zachytit mechaniku a fyziku hry a také to, jak bude zacházet s modely a předměty. A samozřejmě nejdůležitější je zaměřit se na scénář, postavy a svět, ve kterém žijí. Scénář ovlivní umění, mechaniku a prostředí, které je třeba vytvořit.

2.1.3 Produkce

Produkce je nejdelsí a nejdůležitější fází vývoje hry. Zde se utratí lví podíl času a peněz a hra začne skutečně ožívat.

Produkce je místo, kde je zábava, ale je to místo, kde jsou nápady podrobeny konečné zkoušce a musí být učiněna těžká rozhodnutí. V některých případech musí být celé segmenty, mechaniky nebo postavy zcela odstraněny pro dobro projektu..

2.1.4 Testování

V této fázi je nutné důkladně otestovat každý aspekt hry, identifikovat možné chyby, závady, exploity nebo softwarové bloky. Některé z problémů ke kontrole:

Pevné prostředí Schopnost chodit, bojovat nebo se pohybovat po mapě.

Problémy s vykreslováním a výkonem Správné zobrazení objektů a prostředí, zamezení mizení nebo opětovného objevování, stabilita snímkové frekvence.

Zranitelnost Je třeba zabránit tomu, aby byly herní mechanismy používány k získání nespravedlivých výhod.

Měkké zámky Zabraňují náhodnému zaseknutí přehrávače.

Obtížnost Vyvážení úrovně obtížnosti hry.

Skriptové a herecké chyby Kontrola správného pravopisu a použití nejlepších dabingů dabingu.

2.1.5 Před vydáním

Ve fázi před uvedením na trh začíná fungovat marketing. V tuto chvíli by měla existovat stabilní beta verze hry, která může být uvolněna pro veřejnost.

2.1.6 Vydání

Spouštěcí fáze je příležitostí:

1. Opravit všechny zbývající chyby. Studia obvykle uvádějí všechny známé chyby v pořadí podle priority, od chyb, které narušují hru, které ovlivňují stabilitu a výkon, až po drobné kosmetické problémy.
2. Implementovat jednoduchá zlepšení kvality života
3. Přidat finální umělecké úpravy do modelů a prostředí.

2.1.7 Po vydání

Po vydání hry je třeba provést několik kroků:

1. Opravit chyby zjištěné po startu a provést obecnou analýzu..
2. Zvázat přidání obsahu prostřednictvím DLC ke stažení.
3. Poskytovat bezplatné aktualizace obsahu, aby se hráči zajímali.
4. Implementovat záplaty pro vyvážení hry, zvláště pokud jsou po spuštění identifikovány problémy.

2.2 Hádanky a jejich vývoj

Proces tvorby všech her je přibližně podobný, při tvorbě detektivních her však hrají důležitou roli hlavolamy, a to jejich myšlení a tvorba.

Scott Kim ve svém článku [8] dal nejjednodušší definici hádanek: „*Hádanka je zájem a přítomnost správné odpovědi.*“ To je to, co odlišuje hádanky od jiných her – zajímavé, ale bez správné odpovědi, o kterou chce člověk usilovat.

2.2.1 Prvky hádanek

Hádanky mají čtyři hlavní prvky, které je dělají zajímavými: nevyzpytatelnost, přiměřená obtížnost, chyták, dostupnost správné odpovědi.

To znamená, že hráč má maximální požitok z nalezení nového, dosud neznámého řešení, ale aby nevyvolávalo opačné emoce, musí také odpovídat hráčovým schopnostem, zkušenostem a kreativitě. Hráče vždy zajímá hledání skrytého významu ve zdánlivě jednoduchých věcech. Jedním z hlavních prvků dobré hádanky je navíc jasný pocit správnosti řešení poté, co hráč nalezne správnou odpověď.

2.2.2 Kroky návrhu digitálních hádanek

V procesu návrhu digitálních hádanek je klíčové pečlivě projít několika kroky, které zajistí kvalitu a zajímavost výsledné hry. Zde jsou hlavní fáze tohoto procesu:

1. Inspirace
2. Zjednodušení
3. Konstruktor
4. Pravidla
5. Hádanky
6. Testování
7. Stanovení Sekvence
8. Prezentace

Autor knihy *Od konceptu k hratelné hře s Unity a C#* – Jeremy Gibson Bond – v kapitole třináct: Navrhování hádanek velmi jasně popisuje proces tvorby hádanek a uvádí, že inspirace je na samém začátku tvůrčího procesu. Je třeba hledat zdroje inspirace a šikovně omezovat prvky, které tvoří základ budoucí skládačky. Dále následuje zjednodušení. V této fázi je důležité jasně definovat mechaniku hlavolamu, zbavit se zbytečných detailů a zavést jednotnost pro zlepšení struktury. Poté přichází fáze konstrukce, kdy vznikají efektivní nástroje pro vývoj hádanek. Neméně důležitým krokem jsou pravidla. Jejich jasné definování, včetně struktury hrací desky a účelu puzzle, umožňuje konzistentní a logický herní zážitek. Následuje samotný vývoj hlavolamů s různými prvky a mechanikami. Tato fáze poskytuje hráčům různé výzvy a udržuje věci zajímavé. Testování hraje klíčovou roli při hodnocení vytvořených hádanek. Úprava obtížnosti a návrat k předchozím krokům se stává nedílnou součástí procesu. Určení pořadí úrovní je dalším krokem. Rozdělení úrovní od jednoduchých po obtížné krok za krokem zajišťuje plynulou a vzrušující hru. Poslední fází je prezentace. Zde se tvoří vzhled hádanky, rozhraní a zobrazení informací pro hráče jsou dokončeny tak, aby poskytovaly kompletní vizuální vjem [2, str. 261].

2.2.3 Žánry hádanek

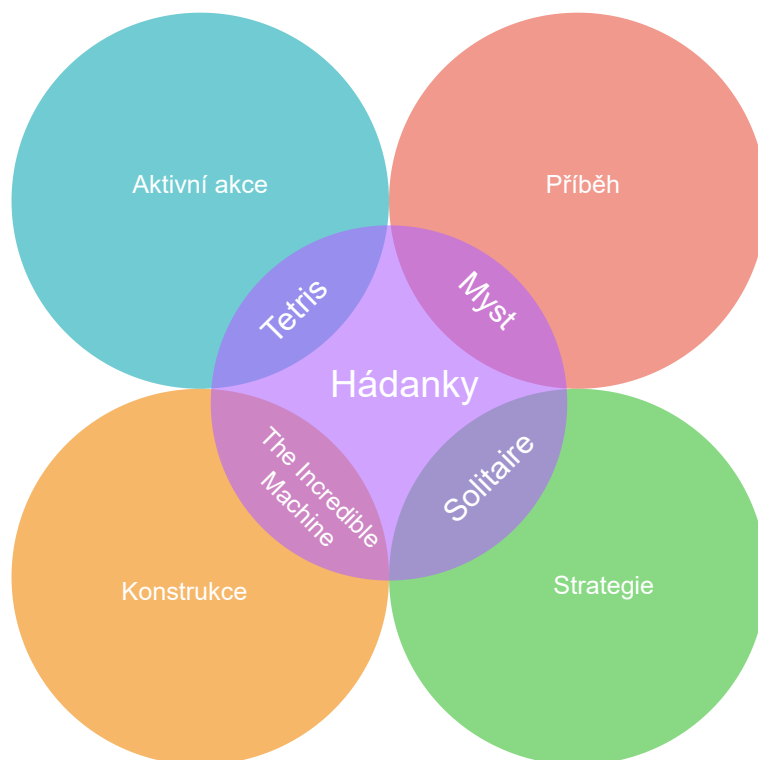
Existují čtyři hlavní žánry hádanek: aktivní akce, příběh, konstrukce, strategie (viz Obrázek 2.2).

Akční hádanky Tento žánr hádanek vyžaduje od hráčů rychlé reakce a koordinaci. Úkoly v těchto hádankách často zahrnují prvky načasování, přesnosti a motoriky. Příkladem mohou být plošinovky s prvky puzzle, kde hráči musí překonávat překážky opatrnými skoky nebo používat předem určené mechanismy k dokončení úrovní.

Dobrodružné hry (příběhové hádanky) Hádanky v dobrodružných hrách často zahrnují řešení hádanek, které posunují příběh. Tyto hádanky mohou po hráčích vyžadovat sbírání předmětů, řešení logických problémů nebo hledání skrytých předmětů. Tyto hry se často zaměřují na průzkum a objevování nových informací, které mohou pomoci rozplést příběh.

Konstrukce (designové hádanky) Tento žánr po hráčích požaduje, aby v daném prostředí vytvořili, postavili nebo navrhli určité objekty nebo struktury. Může se jednat o sandboxové nebo konstrukční simulátory, kde je cílem postavit nejen něco funkčního, ale i efektivního a estetického.

Strategie (strategické hádanky) Hádanky v žánru strategií vyžadují od hráče strategické plánování a myšlení. Hráči musí spravovat zdroje, jednotky nebo postavy, aby dosáhli určitých cílů. V těchto hrách je často nutné předvídat tahy nepřítele a vytvářet dlouhodobé plány, abyste zvítězili.



Obrázek 2.2: Hlavní žánry hádanek. Na obrázku jsou prezentovány čtyři hlavní žánry hádanek: aktivní akce, příběh, design a strategie. Tato vizualizace poskytuje stručný přehled o různorodosti žánrů, což umožňuje hráčům snadný výběr podle svých preferencí a zájmů.

2.3 Druhy detektivních her

Detektivní hry jsou vzrušující žánr, který nejčastěji zahrnuje všechny možné hádanky. Tyto hry však lze rozdělit do několika odrůd, z nichž každá má své vlastní podobnosti a rozdíly, klady a zápory.

Jednou z nejběžnějších forem detektivních her jsou *klasické questy s mechanismem Point-and-Click*. V takových hrách má hráč příležitost prozkoumat svět, sbírat stopy a řešit hádanky, což obecně pomáhá řešit složité hádanky a odhalovat tajemství. Tyto hry často nabízejí vzrušující příběhy, kde každý předmět a akce má svůj význam. Příkladem této rozmanitosti je: *Monkey Island* – tato série klasických questů vytvořených Timem Shaferem

a vyvinutých Lucasfilm Games (později LucasArts). Hráči prozkoumávají barevné ostrovy, řeší hádanky a komunikují s jedinečnými postavami ve stylu komediálního dobrodružství¹.

Interaktivní vyšetřování poskytuje hru, která je více svázána s akcemi hráče. Hráči činí rozhodnutí, která ovlivňují průběh vyšetřování, provádějí výslechy a analyzují důkazy. To vytváří dynamičtější herní zážitek, kde mají hráči pocit, že na jejich výběru skutečně záleží. Příklad toho lze vidět v *L.A. Noire*, který vyvinul Team Bondi a vydal Rockstar Games. Tato hra vyzývá hráče, aby vyšetřovali zločiny v poválečném Los Angeles².

Hororové detektivky – kromě vyšetřování a luštění hlavolamů bývají doplněny o prvky hororu a mystiky. Často zvýrazňují atmosféru strachu a napětí, čímž dodávají vyšetřování další vrstvu složitosti. Klasickým příkladem je hra *Alan Wake* – vytvořená společností Remedy Entertainment, kombinuje prvky detektivky a hororu. Hráči sledují spisovatele Alana Wakea, když prozkoumává záhadné události kolem jeho nejnovějšího románu³.

Tento seznam zahrnuje také *adventury s detektivními prvky*. Kombinují dobrodružné a detektivní prvky a poskytují hráčům širší možnosti. Mohou mít tendenci zkoumat svět volněji, než jen řešit konkrétní hádanky. Příklad: *The Legend of Zelda: Breath of the Wild* – ačkoliv jde především o adventuru, obsahuje i detektivní prvky. Hráč dostane svobodu prozkoumávat otevřený svět, řešit hádanky a objevovat tajemství, aby obnovil paměť hrdiny, Linkovu minulost⁴.

Každý typ detektivní hry má své pro a proti. Úkoly typu Point-and-Click poskytují hluboce pohlcující zážitek, ale mohou působit pomalu. Interaktivní vyšetřování jsou aktivní a poutavá, ale někdy mohou omezit výběr. Hororové detektivky dodávají prvek napětí, ale ne každému vyhovuje hororová atmosféra. Dobrodružné hry s detektivními prvky nabízejí volnější hratelost, ale nemusí se tolik soustředit na vyšetřování.

Výběr mezi těmito odrůdami tedy závisí na preferencích hráče a na tom, které aspekty investigativní zkušenosti jsou pro něj zajímavější. Bylo by proto zajímavé spojit malé části každého typu detektivních her do jedné, aby si každý hráč našel to, co má rád.

2.4 Analýza podobných her

Vzhledem k tomu, že vytvářená hra se zcela nepodobá žádné jiné existující hře, tato část práce představí hry, které inspirovaly vytvoření *The Secret of Summerside*.

2.4.1 Kritérií srovnání

Inspirace k vytvoření této hry přicházela postupně a každá hra byla něčím zajímavá, proto jsou kritéria pro srovnání s podobnými hrami rozdělena do několika typů: Herní mechanika, Grafika a výtvarný styl, Děj a atmosféra.

2.4.2 Herní mechanika

Herní mechanika *The Secret of Summerside* je inspirována hrou *Puzzle Agent*⁵ (viz Obrázek 2.3). Toto je návyková logická hra s vizuálními prvky, ve které hráči následují agenta

¹Odkaz na hru *Monkey Island* v Steamu – https://store.steampowered.com/bundle/6588/Monkey_Island_Collection/

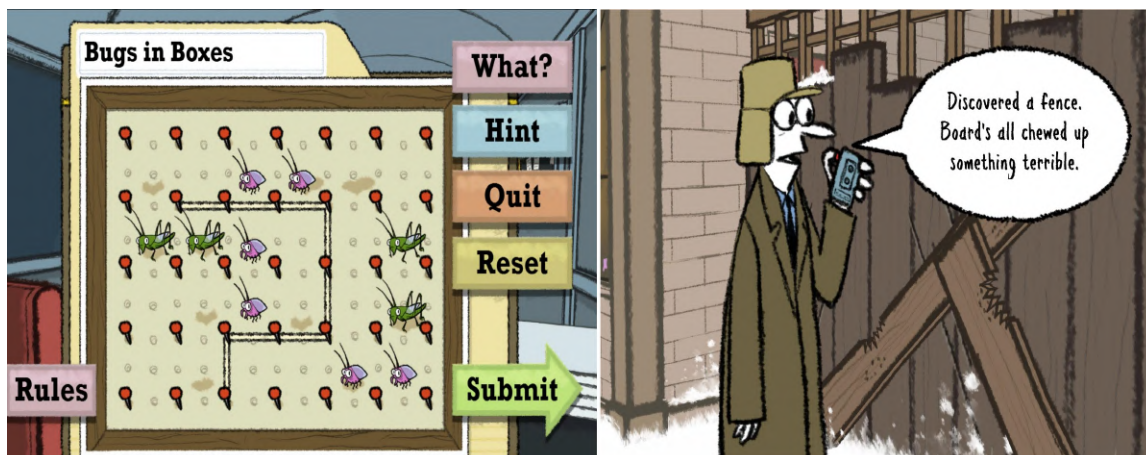
²Odkaz na hru *L.A. Noire* v Steamu – https://store.steampowered.com/app/110800/LA_Noire/

³Odkaz na hru *Alan Wake* v Steamu – https://store.steampowered.com/app/108710/Alan_Wake/

⁴Odkaz na hru *The Legend of Zelda: Breath of the Wild* z Wikipedie – https://en.wikipedia.org/wiki/The_Legend_of_Zelda:_Breath_of_the_Wild

⁵Steam stránky hry *Puzzle Agent* – https://store.steampowered.com/app/31270/Puzzle_Agent/

Nelsona Teatherse z Agentury pro vyšetřování hádanek. Při cestě do tajemného města Scoggins se Tethers setkává s nevysvětlitelnými událostmi, tajemnými obyvateli města a legračními gnómy, to vše je spojeno s hlubokou záhadou. Hráči řeší mnoho hádanek – bludiště, logické problémy, hádanky a mnoho dalšího a odhalují tajemství tohoto světa. Když hráči řeší hádanky, používají vodítka ke sbírání žvýkaček, které pomáhají agentu Tethers přemýšlet a nacházet řešení.



Obrázek 2.3: Snímky ze hry Puzzle Agent. Na prvním obrázku je zobrazena jedna z hádanek, po jejichž vyřešení hráč obdrží odměnu a může se posouvat dále po zápletce. A druhý obrázek ukazuje část vizuální novely, přesně tu hlavní část, která hráče ponoří do děje hry.

Dokonalá souhra mezi vizuálním románem, který hráče vtáhne do příběhu, a poutavými hádankami poskytuje dokonalý základ pro detektivní hru.

2.4.3 Grafika a výtvarný styl

Grafika

Grafika samotné hry, kde jsou implementovány hlavní herní mechanismy, je podobná stylu ze hry *To the Moon*⁶ (viz Obrázek 2.4). Samotný příběh hry se odehrává v neisté budoucnosti a sleduje dva lékaře jménem Dr. Rosalyn a kol. Watt, pracující pro Sigmund Corp. Nabízejí službu pozměňování vzpomínek na umírající lidi, aby mohli zažít život, který nikdy neměli. Hlavním úkolem lékařů je splnit poslední přání klienta.

Vše začíná novou objednávkou od starého muže jménem Johnny. Chce navštívit Měsíc, ale nemůže si vzpomenout proč. Doktoři začnou svou cestu za Johnnyho vzpomínkami, odhalují jeho životní příběh, záhady a tajemství. Prostřednictvím ponoření do různých období a událostí v Johnnyho životě se hráči dozvídají o jeho vztazích, radostech i tragédiích. *To the Moon* zkoumá témata lásky, ztráty a smyslu života.

Grafika ve hře je poměrně jednoduchá, je vyrobena ve stylu *pixel artu*, což jí dodává zvláštní kouzlo a umožňuje hráčům soustředit se na děj a postavy, nikoli na vizuální efekty. Atmosférické pozadí a krásně vykreslené postavy pomáhají zvýraznit emocionální hloubku hry.

⁶Steam stránky hry *To the Moon* – https://store.steampowered.com/app/206440/To_the_Moon/



Obrázek 2.4: Snímky ze hry *To the Moon*. Snímky ukazují zajímavou pixelovou grafiku a umístění kamery jako ortografické projekce, od které se liší pouze úhlem.

Výběr takové grafiky pro detektivní hru byl vědomým rozhodnutím, jehož cílem bylo zjednodušit hráčovo vnímání hádanek a samotných úkolů. Pixel art vytváří jedinečný vizuální zážitek, který umožňuje hráči snadněji se soustředit na detaily a interagovat s herním světem. Tato stylizace slouží nejen jako barevný design, ale také zlepšuje pochopení herních prvků, díky čemuž je proces řešení hádanek příjemnější a přístupnější.

Styl

Styl částí vizuálních románů je zvláštní, je těžké jej plně popsat, je to digitální umění s prvky akvarelové malby, taková grafika trochu připomíná hru *Hollow Knight*⁷ (viz Obrázek 2.5). *Hollow Knight* je hra metroidvania vyvinutá nezávislým studiem Team Cherry.

Hra vypráví příběh Hollow Knighta, malého hrdiny, který se vydává do podzemního světa Hallownest, aby prozkoumal záhadné události zahrnující záhadná stvoření a zapomenuté civilizace. Hlavní hrdina, Hollow Knight, prozkoumává různá místa, bojuje s různými nepřáteli a komunikuje s mnoha postavami. Příběh se vyznačuje hlubokým vyprávěním, které v průběhu hry odhaluje temná tajemství Hollowknighta. Hráč se setkává s tajemnými tvory, skvělými bossy a odhaluje jejich minulost, což dodává zápletce složitost a hloubku. Důležitým prvkem je také průzkum okolního světa a hledání skrytých tajemství.

Grafika v *Hollow Knight* je vyrobena ve stylu metroidvanias s krásnými ručně kreslenými návrhy. Hlavní důraz je kladen na temné a atmosférické dungeony, ponuré lokace a barevné postavy. Vizuální styl vytváří jedinečnou atmosféru, která je vnímána jako tajemná a pohlcující.

⁷Steam stránky hry *Hollow Knight* – https://store.steampowered.com/app/367520/Hollow_Knight/



Obrázek 2.5: Snímek ze hry *Hollow Knight*. Díky grafice a zvláštnímu stylu Metroidvania s krásnými ručně kreslenými ilustracemi lze cítit atmosféru celé hry.

Tento styl je ideální pro detektivní mysteriózní hru, která člověka ponoří do jiného světa.

2.4.4 Děj a atmosféra:

Vytvoření podobné zápletky s mystickou a detektivní atmosférou bylo inspirováno americkým animovaným seriálem *Gravity Falls*⁸ (viz Obrázek 2.6). Dokonce existuje několik her napsaných podle této série.

Děj *Gravity Falls* sleduje dvojčata Dipper a Mabel Pines, kteří jedou strávit léto se svým strýcem Stanem do tajemného městečka Gravity Falls. V tomto malém městě dochází k podivným a neuvěřitelným událostem a dvojčata se rozhodnou tyto záhady prozkoumat.

Jak příběh postupuje, Dipper a Mabel se setkávají s různými nadpřirozenými jevy, setkávají se s výstředními postavami a odhalují tajemství ukrytá v Gravity Falls. Objevují deník, který se stává klíčem k pochopení tajemství města. Jak série postupuje, dvojčata také interagují se strýcem Stanem, který vlastní tajemný Mystery Store.

Gravity Falls není jen o je plná záhad a dobrodružství, ale má také jedinečnou atmosféru, která kombinuje tajemství, humor a emotivní momenty.

⁸Stránka Wikipedie kresleného seriálu *Gravity Falls* – https://en.wikipedia.org/wiki/Gravity_Falls



Obrázek 2.6: Snímek ze kresleného seriálu *Gravity Falls*. Obrázek zobrazuje dvě hlavní postavy animovaného seriálu Gravity Falls, dvojčata Dippera a Mabel, jak běží lesem. V pozadí je viditelná chata podivínů, která patří jejich strýci, a v popředí je místo kmene stromu viditelná noha, která dává tušit záhady tohoto městečka.

Hraní za teenagera je velmi neobvyklým zážitkem, který vám umožní pocíťovat veškerý zmatek a strachy při ovládnutí tohoto světa.

Výsledkem je, že si hotová detektivní hra vypůjčila různé části existujících her a spojila tak nejzajímavější a nejlepší momenty každé hry. Díky směsi mechanik, grafiky a děje se však hra ukázala jako jedinečná a nenapodobitelná.

Kapitola 3

Herní enginy a Unity pro vytváření detektivních her

Tato kapitola obsahuje přehled různých herních enginů a podrobný pohled na Unity Engine s ohledem na jeho využití při vývoji detektivních her. Tato kapitola pojednává o hlavních součástech Unity, jako jsou scény, objekty, komponenty, skripty a zdroje, a zdůrazňuje klíčové funkce, jako je fyzikální modelování, které zajišťuje simulaci chování objektů podle fyzikálních zákonů.

3.1 Herní enginy

Herní engine je softwarové prostředí určené především pro vývoj videoher a obvykle zahrnuje přidružené knihovny a podpůrné programy, jako je editor úrovní. Existuje mnoho herních enginů, z nichž každý je navržen pro konkrétní typ hry nebo cílové publikum. Mezi populární herní enginy patří:

Unreal Engine: Obsahuje výkonný grafický engine, který se široce používá k vytváření vysoce kvalitních 3D her. Poskytuje také mnoho nástrojů pro vizuální programování a skriptování. Díky svému výkonnému grafickému enginu se často používá k vytváření vysokorozpočtových 3D her, včetně stříleček, RGB (*Role-playing game*) her a dobrodružných projektů [4, str. 35].

Godot Engine: Je bezplatný a open source engine se snadno použitelným, vysoce specializovaným programovacím jazykem GDScript. Podporuje 2D i 3D grafiku. Široce se používá v nezávislých projektech a malých hrách, včetně 2D a středně velkých 3D projektů [4, str. 38].

CryEngine: Je známý svou působivou grafickou kvalitou a fotorealistickými scénami. Nabízí širokou škálu nástrojů pro vytváření nejrůznějších her [6, str. 34].

Unity Engine: Je to flexibilní a široce používaný herní engine, který je oblíbený mezi vývojáři nezávislých her, ale používá se i v komerčních projektech. Unity podporuje vývoj jak 2D, tak 3D her a jeho výhodou je intuitivní editor a silné podpůrné nástroje pro vývoj her, včetně podpory pro fyzikální engine, animace a umělou inteligenci. Díky možnosti snadného exportu pro mnoho různých platform, od PC, konzolí až po mobilní zařízení, je Unity ideální volbou pro vývojáře hledající rychlý přechod od prototypu k plnohodnotné hře. Unity je také populární volbou pro vývoj detektivních

her, kde jeho robustní nástroje pro skriptování a podpora pro bohaté narativní prvky umožňují tvůrcům vdechnout život složitým příběhům a hlavolamům [4, str. 33].

3.2 Entity-Component-System (ECS) v herních enginech:

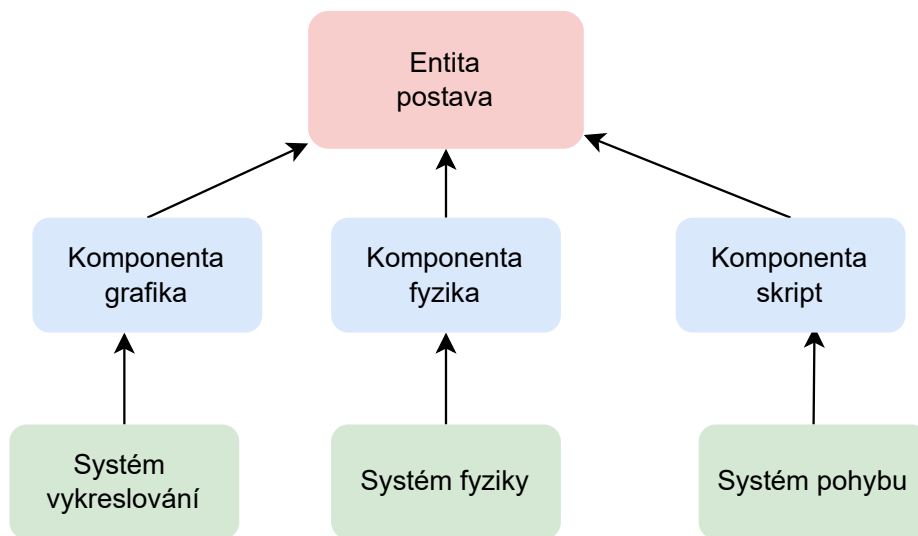
Všechny herní enginey jsou velmi odlišné, ale všechny mají společnou část, která je základem každého engineu – architekturu pro vývoj her.

Entity-Component-System (ECS) je architektonický vzor používaný při vývoji počítačových her pro efektivnější správu objektů v herním světě [12]. Základní myšlenkou ECS je explicitní oddělení dat, logiky a komponent, což umožňuje flexibilnější a efektivnější organizaci kódu (viz Obrázek 3.1).

Entita: Entita představuje objekt v herním světě. Může se jednat o abstraktní reprezentaci jakéhokoli prvku ve hře, ať už jde o postavu, nepřítele, předmět nebo jiný objekt. Každá entita je jedinečná a je definována vlastní sadou komponent.

Komponenta: Komponenty představují data potřebná pro určité aspekty chování entity. Komponenta může například obsahovat informace o grafice, fyzice, zvuku nebo jiných vlastnostech entity. Entita se skládá ze sady komponent, které definují její vlastnosti a chování.

Systém: Systémy představují logiku, která zpracovává komponenty. Provádějí akce s entitami na základě údajů o jejich komponentách. Například systém pro správu fyziky může zpracovávat komponenty odpovědné za fyzikální vlastnosti entity.



Obrázek 3.1: Schéma demonstuje architekturu Entity-Component-System (ECS) používanou při vývoji videoher. Uprostřed je *Entita postava*, která představuje herní objekt, například postavu nebo prvek herního světa. Z tohoto objektu vedou odkazy na tři klíčové komponenty: *Komponenta grafika*, *Komponenta fyzika* a *Komponenta skript*, které odpovídají za vizuální data, fyzické chování a spustitelné skripty. Každá z těchto komponent je propojena s odpovídajícím systémem – *Systém vykreslování*, *Systém fyziky* a *Systém pohybu* – který zpracovává data a chování související s grafikou, fyzikou a pohybem v herním prostoru.

3.3 Výhody Unity pro vytváření detektivních her

Unity je herní engine bohatý na funkce, který se snadno používá a má velkou komunitu vývojářů. Mezi výhody Unity pro vytváření detektivních her patří:

Flexibilita a snadné použití: Unity pracuje s široce používaným programovacím jazykem C#.

Multiplatformní: Unity podporuje více platform, což umožňuje vytvářet detektivní hry pro PC, mobilní zařízení, konzole a virtuální realitu.

Rozmanitost zdrojů: Existuje mnoho hotových zdrojů a pluginů pro Unity, které usnadňují integraci různých prvků detektivních her, jako jsou hádanky, logika vyšetřování a mechanismy hledání.

Vestavěná podpora 2D grafiky a integrace 2D fyziky: Unity má intuitivní nástroje pro 2D grafiku, které usnadňují a zefektivňují vytváření atmosférických pozadí, postav a objektů. Unity má také vestavěnou podporu 2D fyziky, která umožňuje vytvářet realistické chování objektů ve 2D světě, což může být užitečné zejména při vytváření zajímavých hádanek a herních mechanik.

3.3.1 Zásuvné moduly

Unity má velké množství zásuvných modulů, které jsou veřejně dostupné nebo si je lze zakoupit. Tyto pluginy zjednodušují proces vytváření pomocných funkcí pro hru.

Plugin pro správu dialogů:

*Dialogue System for Unity*¹: Tento plugin poskytuje nástroje pro vytváření a správu dialogů postav. Může být použit k vývoji komplexních detektivních scénářů, kde jsou důležité osobní interakce a vývoj zápletky prostřednictvím dialogu.

Zdroje pro vytváření hádanek a hlavolamů:

*Puzzle Game Kit*²: Tento zdroj poskytuje hotové prvky a mechanismy pro vytváření různých hádanek a hlavolamů ve hře. Zahrnuje nástroje pro vývoj investigativní logiky, odhalování klíčových informací a řešení složitých hádanek.

Plugin pro vytváření vyhledávací mechaniky:

*Find Objects System*³: Tento plugin poskytuje možnost vytvářet mechaniky pro vyhledávání objektů ve scéně. V kontextu detektivních her, kde hráč musí najít klíčové stopy, může být tento plugin užitečný pro implementaci aktivních vyhledávacích momentů.

Unity umožňuje vytvářet detektivní hry s jedinečnými hádankami, scénáři a vizuálními styly.

¹ *Dialogový systém pro jednotu* – <https://assetstore.unity.com/packages/tools/ai/dialogue-system-for-unity-11672>

² *Adventure Puzzle Kit* – <https://assetstore.unity.com/packages/templates/systems/adventure-puzzle-kit-175376>

³ *Find It – Hidden Object Template* – <https://assetstore.unity.com/packages/tools/integration/find-objects-advanced-object-detection-system-28172>

3.4 Unity Engine

Unity Engine je v knize o herních vývojářských enginech [6, str. 35] pan Gregory popsán jako „výkonné multiplatformní herní vývojové prostředí podporující širokou škálu platforem“. Tento nástroj umožňuje vyvíjet 2D a 3D hry a také aplikace pro virtuální realitu (VR) a rozšířenou realitu (AR).

Jednou z klíčových vlastností Unity je jeho multiplatformnost, která umožňuje vytvářet aplikace pro různé systémy včetně Windows, macOS, Linux, iOS, Android, WebGL, PlayStation, Xbox, Nintendo Switch a mnoho dalších. Důležitým prvkem vývoje v Unity je programovací jazyk C#, který poskytuje pohodlný a efektivní způsob vytváření herní logiky.

Unity se vyznačuje také aktivní a rozsáhlou komunitou vývojářů. Ta poskytuje snadný přístup k výukovým zdrojům, fórům a podpoře, díky čemuž je proces učení a vývoje v tomto prostředí pohodlnější.

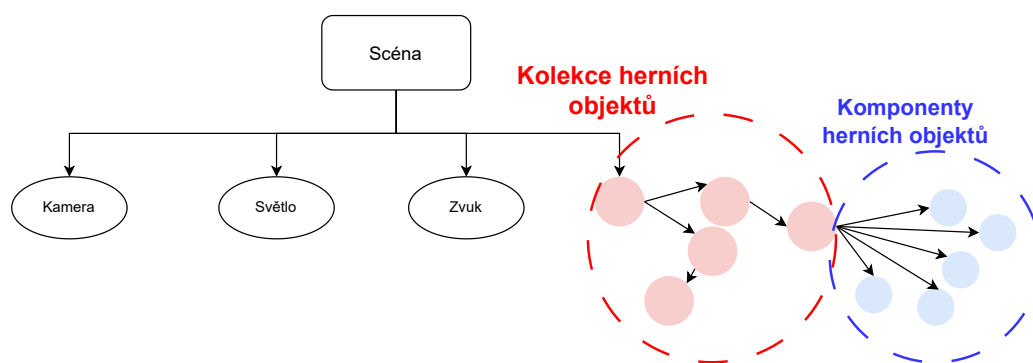
Schopnost Unity efektivně zpracovávat 2D i 3D grafiku z něj činí univerzální nástroj pro širokou škálu projektů. Další prostředky, jako jsou modely, textury a zvuk, jsou snadno dostupné prostřednictvím obchodu Unity Asset Store, což urychluje vývoj.

V neposlední řadě poskytuje Unity vestavěné nástroje pro tvorbu obsahu ve virtuální a rozšířené realitě, což z něj činí oblíbenou volbu v těchto inovativních oblastech.

Tato část popisuje součásti Unity, které se používá k vytvoření hry (viz Kapitulu Implementace) : scény, entity, komponenty, skripty, zdroje, fyzikální systém a prefaby. Veškerý popis částí byl přebrán z oficiální Unity Engine dokumentace [14].

3.4.1 Scény

Unity umožňuje vytvářet scény podobné scénám ve filmu. Manuál uvádí: „Ve scénách pracujete s obsahem v Unity. Jsou to kolekce, které obsahují celou hru nebo aplikaci nebo její části.“⁴ Pod scénou si lze představit: hlavní nabídku hry (viz Obrázek 3.4), nastavení, úroveň a další.

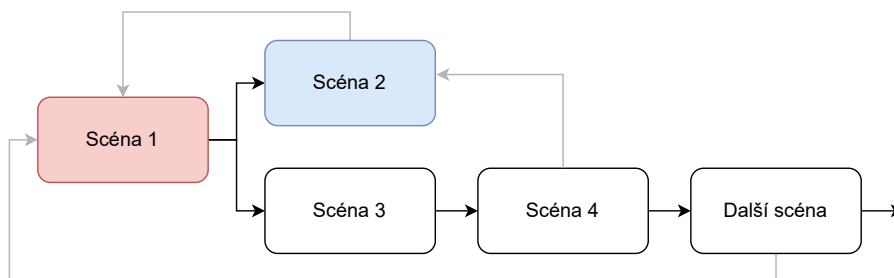


Obrázek 3.2: Tento diagram ukazuje strukturu herní scény v Unity, včetně klíčových komponent, jako je kamera, zdroje světla a zvuku a kolekce herních objektů. Kromě toho je třeba poznamenat, že herní objekty, které tvoří scénu, jsou zase komplexní entity sestávající z více komponent, z nichž každá je zodpovědná za určité aspekty chování a funkčnosti objektu v herním světě. Tyto prvky jsou nedílnou součástí tvorby scény a ovlivňují vnímání a interakci uživatele s herním světem.

⁴ Unity manual, scény – <https://docs.unity3d.com/Manual/CreatingScenes.html>

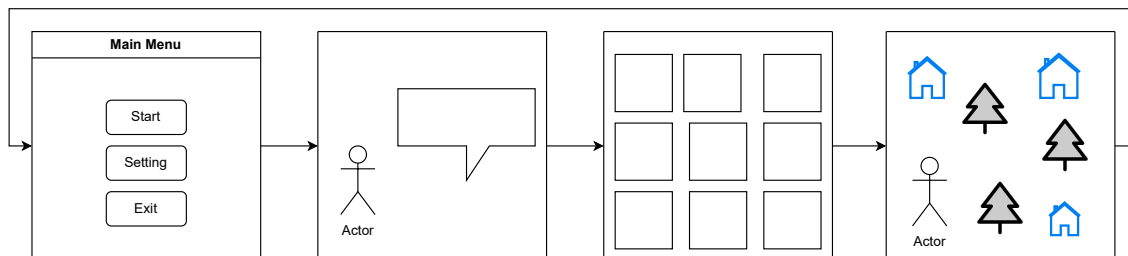
Scéna v Unity je soubor objektů a komponent spojených do jednoho vizualizačního a interakčního prostředí. Zahrnuje také prvky potřebné k vytvoření kompletního herního světa, jako je kamera, světla a zvukové efekty (viz Obrázek 3.2).

Klíčovým aspektem scén v Unity je možnost jejich přepínání (viz Obrázek 3.3). To znamená, že vývojáři mohou vytvářet samostatné scény pro různé části hry, například pro hlavní nabídku, nastavení, herní úrovně atd.



Obrázek 3.3: Schéma zobrazující mechanismus přepínání mezi různými scénami v Unity. Demonstruje různé možnosti navigace, které má uživatel k dispozici. Je možné se vrátit k předchozím scénám a poté pokračovat ve hře dále.

V Unity může být každá scéna jedinečným světem obsahujícím zcela odlišné objekty, jako jsou postavy, krajiny, zdroje světla a zvuku. To umožňuje vývojářům pohodlněji organizovat a spravovat různé úrovně a části hry, vytvářet různé herní scénáře a zlepšovat hrátelnost. To umožňuje vývojáři přizpůsobit jakoukoli scénu pro speciální účely, které jsou funkčně odlišné, například: hlavní menu, nastavení a jednotlivé herní úrovně.



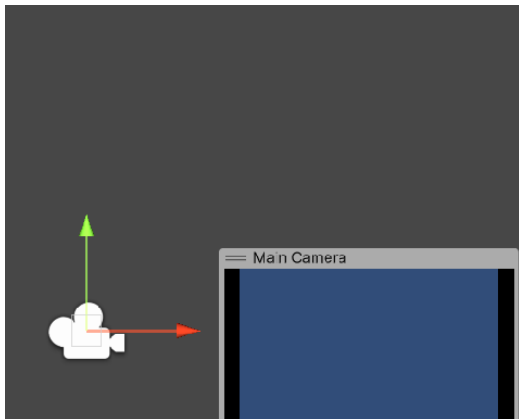
Obrázek 3.4: Schéma ukazuje proces přepínání mezi scénami v Unity, počínaje hlavní nabídkou, která obsahuje možnosti *Start*, *Nastavení* a *Ukončit*. Schéma ukazuje, jak mohou vývojáři efektivně řídit přechody mezi scénami, a zajistit tak plynulý herní zážitek a rozmanitý herní obsah.

Přepínání mezi scénami umožňuje vytvořit z každé části hry samostatný prvek, což výrazně usnadňuje správu a aktualizaci obsahu. Hráč může například začít s hlavní nabídkou, pak se přepnout do nastavení, pak přejít do herních úrovní atd. a vytvořit tak plynulý a nepřetržitý tok hry.

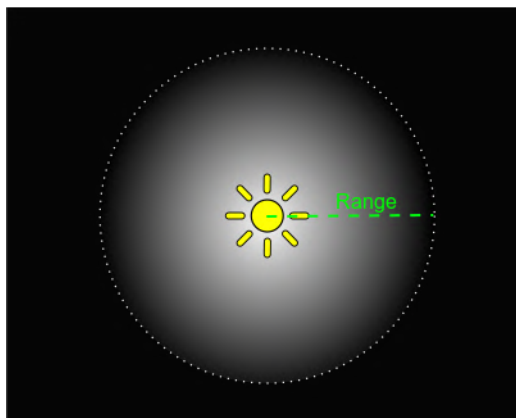
Scény v Unity tak představují mocný nástroj pro organizaci a strukturování herního obsahu, který poskytuje flexibilitu a pohodlí při vývoji her.

3.4.2 Entity

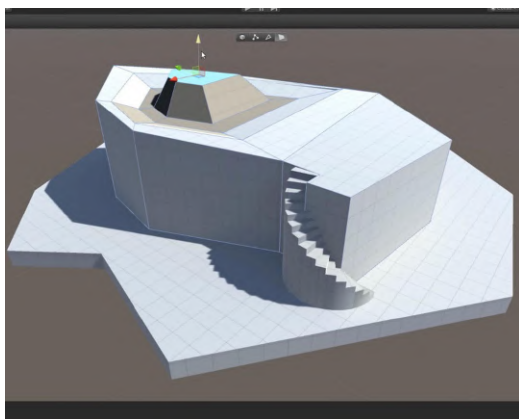
V Unity se scény skládají z objektů a každý objekt (viz Obrázek 3.5) se skládá z komponent. Manuál uvádí: „Každý objekt ve hře je *GameObject*, od postav a předmětů až po světla, kamery a speciální efekty.“⁵ Předměty mohou být postavy, předměty, budovy atd. Tyto předměty se mohou vzájemně ovlivňovat a vytvářet herní mechaniky.



(a) Objekt v Unity – Kamera: Kamera v Unity definuje úhel pohledu hráče, skrze který hráč vidí herní svět.



(b) Objekt v Unity – Světlo: Světlo v Unity vytváří atmosféru scény tím, že jí dodává hloubku a realističnost prostřednictvím osvětlení objektů.



(c) Objekt v Unity – 3D model: 3D model v Unity slouží k vytvoření trojrozměrných postav, objektů a prostředí ve hře.



(d) Objekt v Unity – 2D sprity: 2D sprity v Unity se používají k vytváření postav a prvků prostředí ve 2D hrách.

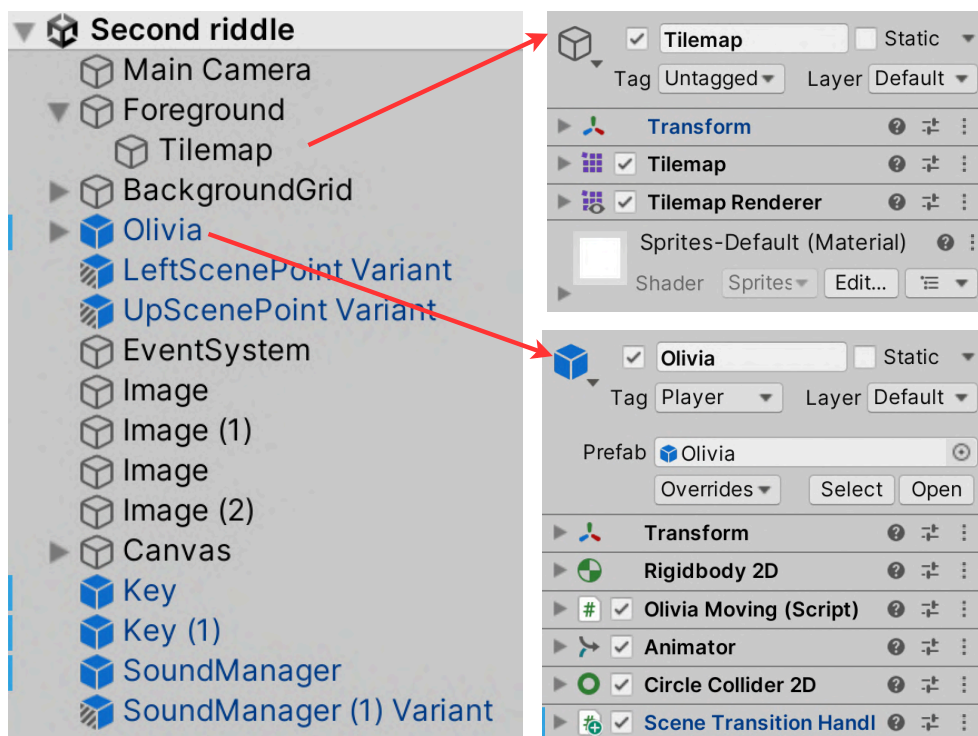
Obrázek 3.5: Příklady čtyř různých herních objektů.

3.4.3 Komponenty

Každý objekt v Unity se skládá z komponent (viz Obrázek 3.6). Manuál uvádí: „Komponenty jsou funkční části každého *GameObject*. Komponenty obsahují vlastnosti, které můžete

⁵ Unity manual, objekty – <https://docs.unity3d.com/Manual/GameObjects.html>

upravit a definovat tak chování objektu *GameObject*.“⁶ Komponenty jsou jakési instrukce, které říkají objektu, jak se má chovat. Kromě polohy, rotace a měřítka mohou obsahovat grafiku, kolizní prvky pro řešení kolizí, skripty s programovým kódem, zdroje zvuku, světla, animátory pro ovládání animací a fyzické komponenty pro interakci s fyzickým světem. Postava může mít například komponent, který jí říká, aby se pohybovala nebo střílela.



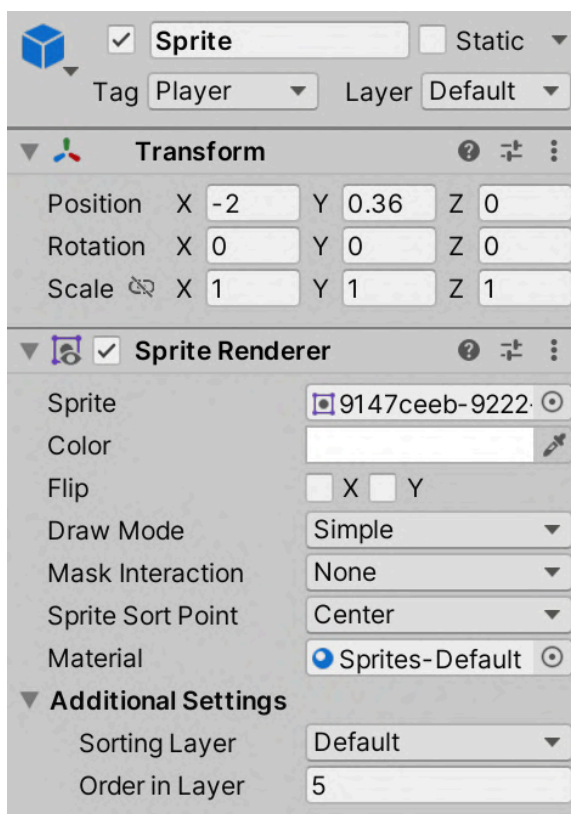
Obrázek 3.6: Diagram znázorňuje strukturu scény, která se skládá z různých herních objektů, jež tvoří stromovou hierarchickou strukturu známou jako graf scény. Tato struktura umožňuje spravovat vztahy a závislosti mezi objekty a zajišťuje jejich logické uspořádání. Objekty mohou například dědit transformace od svých nadřazených prvků. Každý objekt ve scéně se skládá z komponent, jak ukazují příklady na pravé straně diagramu. Tyto komponenty přidávají objektům různé vlastnosti a chování, například možnost zobrazení na scéně, fyzickou interaktivitu nebo specifické herní mechaniky. Příkladem takové kompozice je objekt *Tilemap*, který se skládá z komponent zodpovědných za vykreslování mapy dlaždic a dalších komponent, jako jsou kolizní prvky, které umožňují interakci s objekty na mapě. Podobně může objekt *Olivia* obsahovat komponenty, které definují její chování – *skript* a interakci s ostatními objekty v herním světě – *fyzické komponenty*.

V Unity jsou objekty ve scéně uspořádány hierarchicky, přičemž každý objekt může obsahovat podřízené objekty a komponenty, které určují jeho chování a vzhled. Každý objekt může mít vlastní kolekci komponent, které daný objekt ovládají. Například na obrázku 3.6 vidíme objekt *Tilemap*, který má komponenty *Transform*, *Tilemap* a *Tilemap Renderer* – ty definují pozici, interaktivní mřížku, respektive vizuální reprezentaci dlaždic.

V Unity má každá komponenta mnoho parametrů, které lze individuálně přizpůsobit potřebám. Například komponenta transformace, která řídí polohu, natočení a měřítko objektu v trojrozměrném prostoru nebo komponenta vykreslování spritu, která umožňuje vybrat ob-

⁶ Unity manual, komponenty – <https://docs.unity3d.com/Manual/Components.html>

rázek spritu, jeho barvu a další vizuální vlastnosti, jako je režim zobrazení a interakce s maskami (viz Obrázek 3.7). Každý parametr těchto komponent je modifikovatelný, což vývojářům poskytuje flexibilní nástroje pro přizpůsobení vzhledu a umístění objektů ve hře nebo aplikaci. Tento systém komponent je základem pro vytváření a správu prvků herního světa.



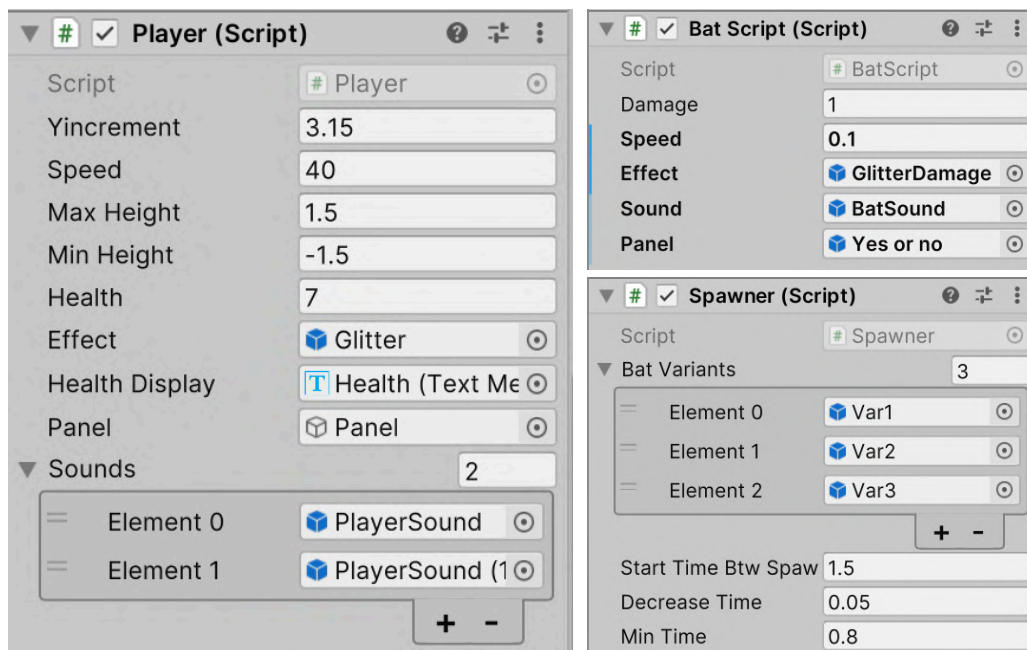
Obrázek 3.7: Na zobrazeném obrázku má uživatel možnost přizpůsobit prvky prostřednictvím grafického rozhraní, včetně jejich umístění a vizuálních aspektů. Tyto prvky, které mají parametry umístění a vnějšího stylu, lze přizpůsobit tak, aby bylo v projektu dosaženo požadovaného efektu.

3.4.4 Skripty

Jednou z komponent objektu může být skript s programovým kódem. Unity používá k psaní skriptů programovací jazyk C#. Skripty jsou programy, pomocí kterých objekty provádějí různé akce. Jejich přidáním jako komponenty k určitému objektu (viz Obrázek 3.8) lze ovládat jeho chování, například objeovávání, mizení, přesouvání atd.

Skripty mohou být velmi rozmanité a plnit mnoho úkolů: od ovládání pohybu postav až po vytváření složitých interakcí v herním světě a realizaci herní logiky.

Skripty se používají k animaci statických objektů na scéně, například k tomu, aby herní postava měla možnost pohybu nebo interakce s okolním světem. Skript může být zodpovědný za sledování zdraví postavy, ovládání umělé inteligence protivníků, počítání hráčských bodů, ovládání herního rozhraní a mnoho dalšího.



Obrázek 3.8: Na prvním obrázku je zobrazen skript *Player* pro objekt hráče, včetně parametrů jako *Yincrement* (vertikální přírůstek), *Speed* (rychlost), *Max Height* (maximální výška) a *Min Height* (minimální výška), které určují pohyb a základní vlastnosti postavy ve hře. Dále jsou zde zobrazeny parametry *Health* a *Effect*, které ovlivňují interakci postavy s herním světem. Druhý snímek obrazovky ukazuje skript *Bat*, který nastavuje parametry dalšího herního objektu, netopýra. Mezi tyto parametry patří Poškození, Rychlost a Efekt (efekt *GlitterDamage*), které určují chování tohoto objektu ve hře. Dále je zde snímek skriptu *Spawner*, který řídí spawnování netopýrů, s parametry *Bat Variants*, *Start Time Btw Spaw*, *Decrease Time* a *Min Time*. Oba skripty používají atribut *SerializeField* [9, str. 17], který zviditelňuje proměnné v inspektoru Unity a umožňuje vývojářům snadno měnit klíčové hodnoty, aniž by museli měnit zdrojový kód.

Skripty v Unity se aktivují přidáním jako komponenty k objektům scény, což umožňuje objektům reagovat na vstupy uživatele, jako jsou stisky kláves nebo kliknutí myši. Tyto softwarové moduly mohou naslouchat vstupním událostem a zpracovávat je, spouštět příslušné funkce v reakci na akce uživatele, a vytvářet tak interaktivní a dynamické prostředí. Kromě toho mohou skripty interagovat s dalšími komponentami a objekty ve scéně, což umožňuje komplexní řízení chování při hře Více o životním cyklu skriptů se píše v kapitole 3.5.

3.4.5 Zdroje

Zdroje (*Assets*), viz Obrázek 3.9 (Obrázek převzatý⁷), jsou vše, co se ve hře používá: obrázky, zvuky, modely a mnoho dalšího. Unity usnadňuje import a používání prostředků, aby hra získala jedinečný styl.

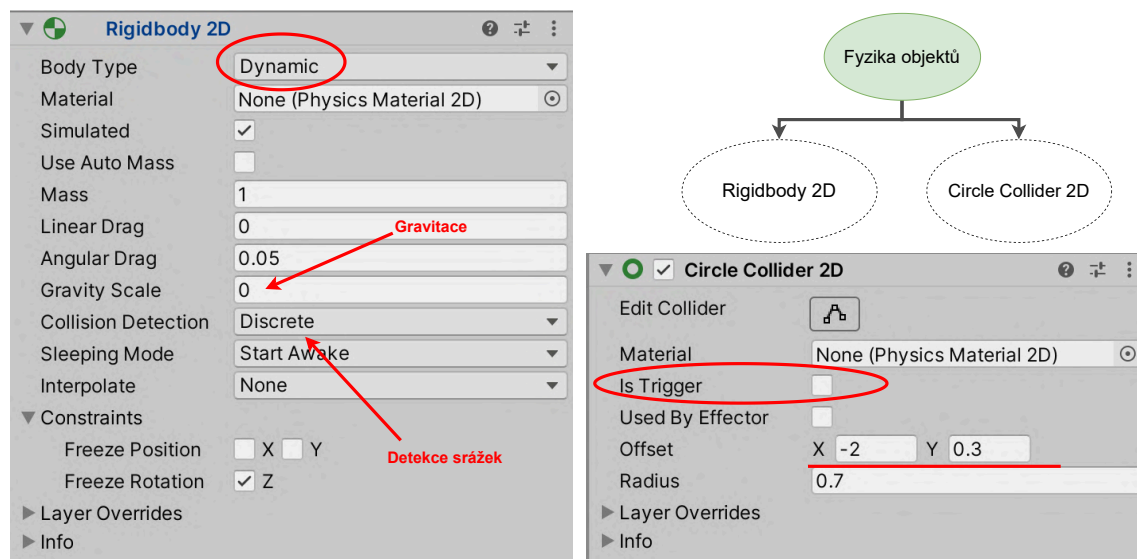
⁷Obrázek – <https://docs.unity.cn/ru/2021.1/Manual/GameObjects.html>



Obrázek 3.9: Různé herní zdroje, zejména sprity: 3D postavy, světla, 3D model stromu, zvuk atd.

3.4.6 Systém pro správu fyziky

Manuál uvádí: „Unity pomáhá simulovat fyziku v projektu, aby bylo zajištěno, že objekty správně zrychlují a reagují na srážky, gravitace a různé další síly.“⁸ Unity obsahuje fyzikální engine (viz Obrázek 3.10), který umožňuje objektům komunikovat s okolní 3D scénou podle fyzikálních zákonů pevných těles. Může to být gravitace, srážky, třštění a další.



Obrázek 3.10: První obrázek ukazuje konfiguraci komponenty *Rigidbody 2D* pro herní objekt ve 2D herním enginu, jako je Unity. Nastavené parametry naznačují, že objekt bude dynamický (bude podléhat fyzice), ale gravitace je vypnutá (Gravity Scale je 0). To znamená, že objekt nebude podléhat vlivu gravitace, což lze využít například pro objekty, které se potřebují vznášet ve vzduchu nebo se pohybovat ve stavu beztíže. Zároveň objekt nemůže procházet jinými objekty a bude s nimi interagovat podle stanovených fyzikálních pravidel (Detekce srážek je nastavena jako Diskrétní, což zajišťuje detekci srážek). Druhý obrázek ukazuje komponentu *Circle Collider 2D*, která definuje kruhovou kolizní zónu kolem objektu. Posunutí (Offset $[-2, 0.3]$) mění polohu kolizní zóny vzhledem ke středu objektu, což lze využít k vytvoření asymetrických objektů nebo k posunu těžiště. Protože parametr *Is Trigger* není zaškrtnutý, kolider bude interagovat s ostatními fyzickými objekty ve hře a zabrání jejich průchodu.

⁸ Unity manual, fyzika – <https://docs.unity3d.com/Manual/PhysicsSection.html>

Konfigurace komponent *Rigidbody 2D* a *Circle Collider 2D* v herním enginu, jako je Unity, umožňuje vytvořit mnoho různých efektů změnou jejich parametrů:

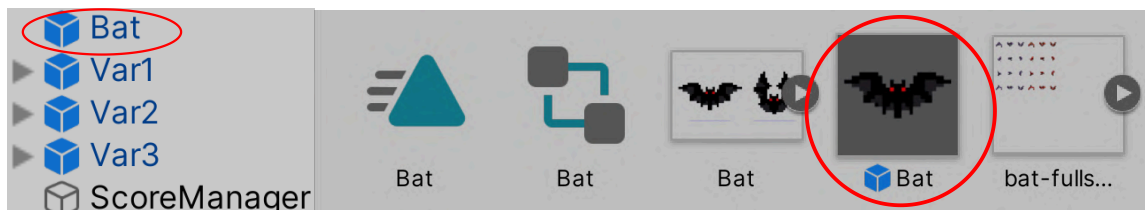
- **Dynamické chování:** Změna parametru *Body Type* na *Kinematic* nebo *Static* změní způsob, jakým objekt reaguje na fyziku a kolize. Kinematické objekty se mohou pod kontrolou kódu pohybovat, ale nereagují na síly a kolize, zatímco statické objekty se nepohybují vůbec a mohou sloužit jako nehybné překážky.
- **Simulovaná hmotnost:** Aktivace možnosti *Use Auto Mass* umožní systému automaticky vypočítat hmotnost na základě velikosti a hustoty kolidujícího objektu. Pokud je tato možnost vypnutá, můžete hmotnost objektu nastavit ručně, což ovlivní jeho reakci na kolize a síly.
- **Stupnice gravitace:** Zvýšení parametru *Gravity Scale* způsobí, že objekt bude padat rychleji, jako by se nacházel v prostředí s vyšší gravitací.
- **Kolizní zóna:** Změnou parametrů *Offset* a *Radius* v *Circle Collider 2D* lze vytvořit efekty, jako je kolísající rovnováha objektu nebo nestandardní chování při kolizi, což může být užitečné pro vytvoření jedinečné herní mechaniky.
- **Spouštěče:** Povolením možnosti *Is Trigger* (Je spouštěč) se kolizní prvek změní na spouštěč, který nebrání průchodu jiných objektů, ale může spouštět události nebo skripty, když s nimi zkříží cestu.
- **Spánek a probuzení:** *Sleeping Mode* řídí optimalizaci výkonu tím, že určuje, kdy má objekt přestat být simulován, dokud nenastane událost, která jej *probudí*.
- **Omezení:** Pomocí sekce *Constraints* lze zmrazit polohu nebo natočení objektu v určitých osách, což je užitečné při vytváření plošin, které se pohybují pouze jedním směrem, nebo objektů, které by se neměly otáčet.

Změnou těchto parametrů můžete vytvářet objekty s nejrůznějšími fyzikálními vlastnostmi a chováním, což otevírá širokou škálu možností pro herní nápady.

3.4.7 Prefaby

Prefaby v Unity představují mocný nástroj pro opakované použití konfigurace herních objektů, včetně jejich komponent, nastavení a vnořených objektů. Vytvořením Prefabu vývojář ukládá prototyp objektu, který lze opakovaně umisťovat do scén nebo instancovat během běhu hry. To je obzvláště užitečné pro prvky, které se v hře často používají, jako jsou postavy, nepřátelé, střely, mince a další herní elementy.

Používání Prefabů zvyšuje efektivitu vývoje a zjednodušuje proces provádění změn. Pokud je třeba změnit vlastnost objektu, lze to udělat jednou v Prefabu, a všechny instance tohoto Prefabu se automaticky aktualizují ve hře. To umožňuje vývojářům snadno provádět globální změny a udržovat konzistentnost herních prvků na všech úrovních.



Obrázek 3.11: Na obrázku jsou dva objekty ilustrující příklady prefabů v Unity. První objekt je jednoduchý herní prvek – *Bat*, zatímco druhý objekt je složitější konstrukce, kombinace několika prvků. Oba tyto objekty mohou vývojáři používat k vytváření herních světů, což usnadňuje umísťování a opakované používání již přizpůsobených prvků v různých scénách a herních úrovních. Vlevo jsou objekty v okně hierarchie scény, které se přenesením do složky projektu staly prefabami jak je zobrazeno vpravo.

Kromě toho Prefaby podporují koncept *varianty* (variants), což umožňuje vytvářet různé verze téhož objektu s unikátními charakteristikami, aniž by přitom ztratily spojení s původním Prefabem. To zjednodušuje správu několika podobných objektů, které mají drobné rozdíly.

Pro vytvoření Prefabu musí vývojář přetáhnout objekt z okna hierarchie scény do složky projektu v editoru Unity. Poté se objekt stane Prefabem (viz Obrázek 3.11) a může být umístěn na jakoukoliv scénu nebo vytvořen dynamicky prostřednictvím skriptů.

3.5 Životní cyklus projektu v Unity

V herním enginu Unity se projekty se řídí pomocí smyčky znázorněné na obrázku 3.12. V Unity má každý vývojář možnost psát skripty, které řídí životní cyklus projektu a umožňují skriptům reagovat na akce hráče nebo se spouštět v určitý čas. Pomocí metod, jako je *Update* pro průběžnou kontrolu podmínek nebo *Start* a *Awake* pro inicializaci, je možné ve hře vytvářet složité interakce. To dává vývojářům mocný nástroj k vytváření dynamických a reaktivních herních světů, kde každá akce hráče může vyvolat specifické zpětné volání, což činí hraní zajímavějším. Zde je uvedeno, jak fungují jednotlivé fáze životního cyklu skriptu⁹.

Inicializace

- Probuzení – Zavoláno po načtení skriptu.
- Spuštění – Zavolán před prvním snímkem, pokud je skript aktivní.

Redaktor

- Obnovení – V editoru se volá, když je skript resetován nebo připojen k objektu.

Fyzika

- FixedUpdate – Provádí se každý pevně stanovený interval a slouží k aktualizaci fyziky.
- Interní aktualizace fyziky – Volá se po aktualizaci každého objektu ve scéně. Používá se k aktualizaci pozic a stavů objektů po dokončení všech volání aktualizace.

⁹https://www.youtube.com/watch?v=L_Wd4emmcLQ

Vstupní události

- OnMouseXXX – Události spouštěny různými akcemi myši.

Herní logika

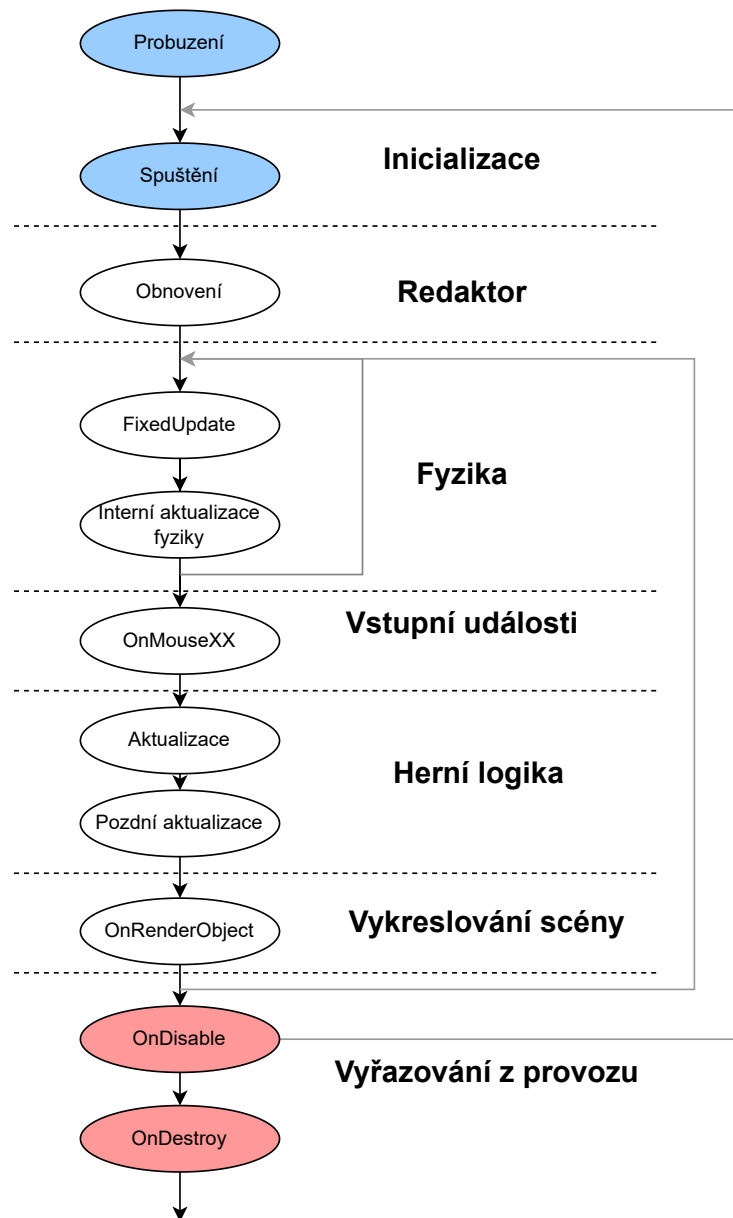
- Aktualizace – Volána každý snímek a slouží k běžné aktualizaci herní logiky.
- Pozdní aktualizace – Volá se pro aktualizace, které by měly proběhnout po hlavní aktualizaci hry.

Vykreslování scény

- OnRenderObject – Události související s vykreslováním a volané v určitém pořadí.

Vyřazení z provozu

- OnDisable – Volá, když se skript stane neaktivním.
- OnDestroy – Voláno, když je skript nebo objekt zničen.

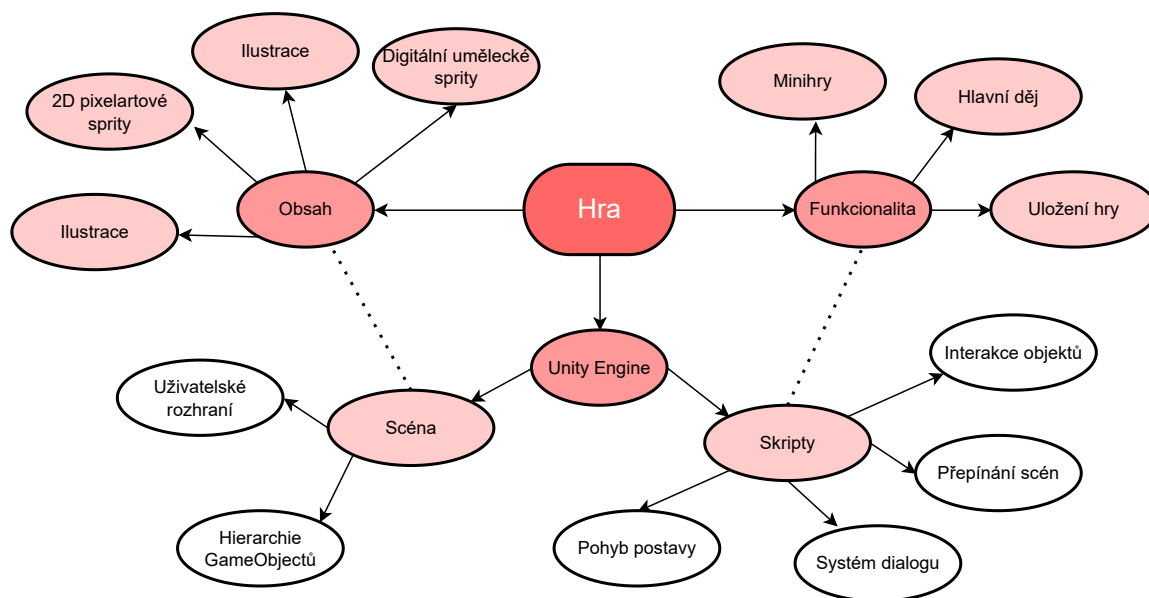


Obrázek 3.12: Životní cyklus skriptu v Unity. Smyčka představující diagram životního cyklu skriptu.

Kapitola 4

Návrh, koncept a logická struktura hry

Tato kapitola popisuje návrh aplikace, její distribuci a funkčnost. Hlavní cíl hry dokončit celý děj a všechny minihry a dostat se na konec příběhu. Na obrázku 4.1 je znázorněno rozdělení hry na části, kde je popsána její funkčnost v této kapitole, a obsah a implementace pomocí engine Unity je popsána v kapitole 5.



Obrázek 4.1: Výslednou hru lze rozdělit na tři hlavní části: obsah, funkčnost a implementaci v engine Unity. Obsah je vše, co hráč ve výsledné aplikaci vnímá v podobě vizuálních informací. Pro hru byl vytvořen podrobný příběh. Na základě tohoto příběhu byly vytvořeny ilustrace, sprity a modely. Herní logiku a pravidla všech miniher i hlavní hry lze zakomponovat do funkčnosti aplikace. Vše je realizováno pomocí engine Unity. Funkčnost je realizována pomocí skriptů a interakce objektů ve scénách. Veškerý obsah se zobrazuje pomocí scén.

4.1 Průběh hry

Hra je strukturována do několika fází, přičemž každá fáze reprezentuje klíčový segment herního dobrodružství. Tato struktura je graficky znázorněna na obrázku 4.2, který podrobně rozkresluje průběh hry od úvodního segmentu až po závěrečnou bitvu. Každá fáze obsahuje své vlastní herní mechanismy, úkoly a výzvy, které hráč musí splnit pro postup dál.

Úvod: Hra začíná úvodním segmentem, ve kterém je hráčům prostřednictvím krátkého úvodního videa a textu představen kontext odjezdu hlavního hrdiny do mystické vesnice na léto (viz Obrázek 4.2). Hlavním účelem tohoto segmentu je seznámit hráče s příběhem a motivací postavy a představit hlavní cíl hry – vyřešení záhady kouzelné studny.

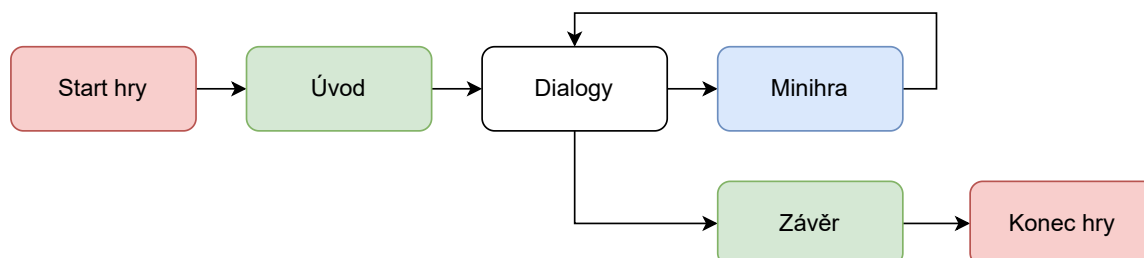
Sestavení mapy: Dále hra přechází do fáze sestavování mapy. Hráč bude muset z rozbitých kousků mapy sestavit skládačku a poté věnovat čas zapamatování si tras a klíčových bodů na mapě.

Procházení labyrintem: Po sestavení mapy začíná fáze překonávání labyrintu domů a lesů, ve kterém se hráč musí orientovat s pomocí sestavené mapy. V této části hry hraje důležitou roli zapamatování si tras a klíčových bodů bludiště, stejně jako sbírání potřebných klíčů pro přístup k dalším úsekům trasy.

Najdi dvojici: Po dosažení studny čeká hráče úkol *Najdi dvojici*, kdy musí přiřadit odpovídající klíč k zámku na studni. Tento prvek hry vyžaduje pozornost a logiku, protože správná volba klíče umožní pokračovat v cestě.

Setkání s trpaslíkem: Poté, co hlavní hrdina sestoupí do studny a zjistí, že je prázdná, potká kouzelného trpaslíka, který mu odhalí skutečnou historii vesnice. Tento okamžik je klíčovým zvratem v ději a představuje nový cíl – porazit zlého čaroděje.

Tajný tunel a závěrečný boss: Závěrečná fáze hry zahrnuje cestu tajným tunelem, vyhýbání se setkání s magickými netopýry a závěrečnou bitvu s čarodějem, který drží všechny vesničany ve strachu. Úspěch v této bitvě nejen osvobodí vesnici od zla, ale také odhalí všechna tajemství herního světa.

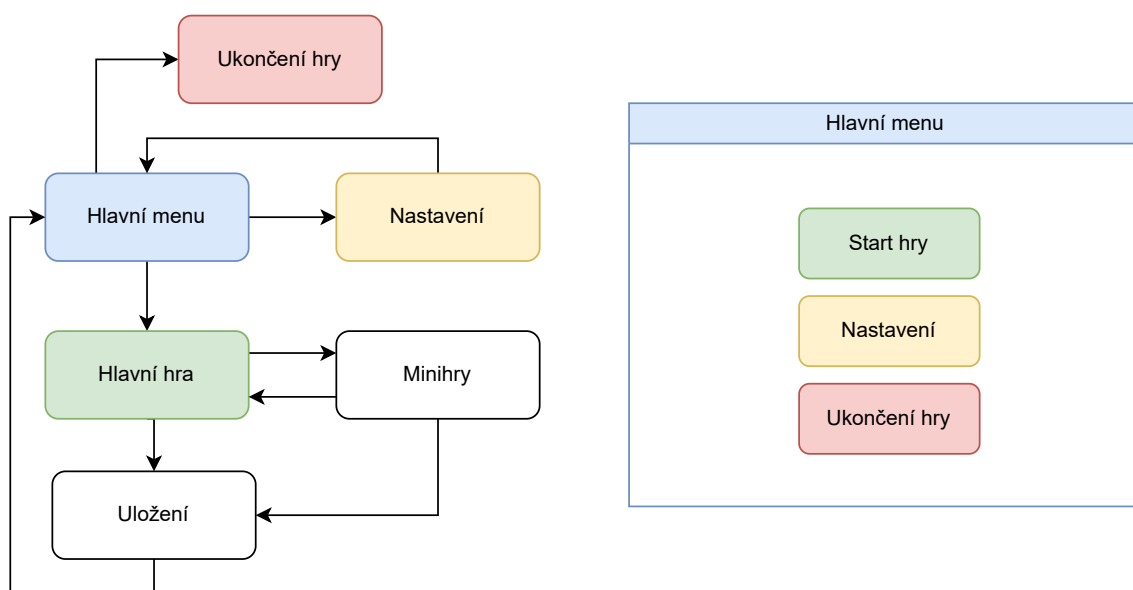


Obrázek 4.2: Hra je zahájena fází *Start hry*, která je vstupním bodem a určuje kontext pro další vývoj událostí. Fáze *Úvod* slouží k seznámení uživatele se základními mechanismy a příběhovým pozadím a poskytuje základ pro další interakce. Postupem do fáze *Dialogy* se spouští systém interakce s NPC (nehráčskými postavami), což umožňuje hlubší ponor do příběhu. *Minihra* zavádí další herní prvek, který vyžaduje, aby hráč pro svůj postup použil specifické dovednosti nebo vyřešil úkoly. Herní cyklus je ukončen ve fázi *Závěr*, která přechází do fáze *Konec hry*, kde jsou shrnuty a vyhodnoceny výsledky herního zážitku.

4.2 Úvodní menu

Úvodní menu hry je hlavním uživatelským rozhraním, které hráče přivítá po spuštění hry (viz Obrázek 4.3). Je důležitou součástí herního zážitku, protože slouží jako vstupní brána ke všem hlavním funkcím hry. V této kapitole jsou popsány prvky, které tvoří hlavní nabídku, jejich vzájemné vztahy a funkce.

Výběrem možnosti *Spustit hru* lze zahájit novou herní relaci nebo obnovit poslední uloženou hru. Sekce nastavení umožňuje hráčům přizpůsobit si herní zážitek podle svých preferencí. Lze zde přizpůsobit jazyk. Minihry jsou samostatný herní obsah, který nabízí jedinečné výzvy a zábavu mimo hlavní příběh hry. Funkce ukládání hry má zásadní význam pro to, aby hráči měli možnost uložit si svůj postup. Možnost Dokončení hry poskytuje bezpečné ukončení hry uložením veškerého aktuálního postupu. To zajišťuje, že se hráči mohou do hry kdykoli vrátit, aniž by přišli o dosažené úspěchy.



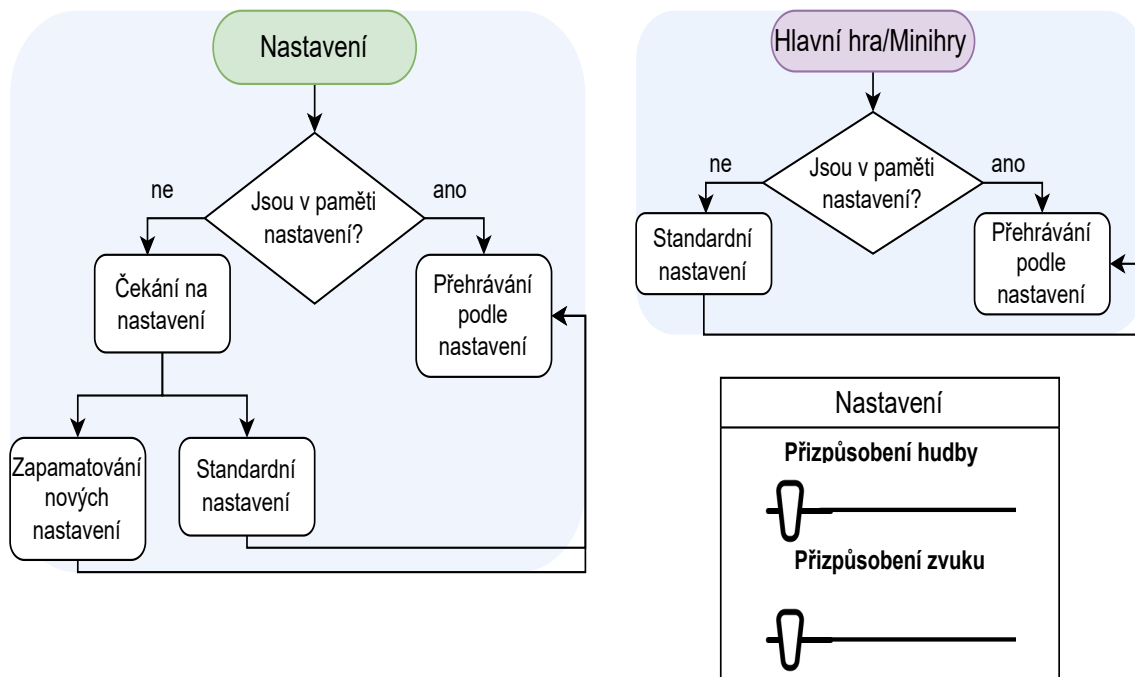
Obrázek 4.3: Obrázek ukazuje hlavní navigační systém hry, který je rozdělen na dvě hlavní části: funkční a strukturální. Funkční část, zobrazená vlevo, zobrazuje klíčové ovládací prvky hry, jako je přepínání do hlavní hry, ukládání postupu a výběr miniher, a odráží nejdůležitější aspekty hraní. Strukturální část, zobrazená vpravo, představuje *wireframe* (rozvržení) hlavní nabídky, včetně dalších možností: spuštění hry, nastavení a ukončení hry.

4.3 Nastavení

Nastavení zvuku v naší hře umožňuje uživatelům přizpůsobit hlasitost hudby na pozadí a zvukových efektů. Tato nastavení jsou uložena v paměti hry a jsou použita ve všech úrovních, což zajišťuje, že zvuk je vždy konzistentní v průběhu hry.

Rozhraní nastavení zvuku se skládá ze dvou posuvníků (viz Obrázek 4.4), z nichž jeden slouží k nastavení hlasitosti hudby na pozadí a druhý k nastavení hlasitosti zvukových efektů. Uživatel může měnit polohu každého posuvníku, aby dosáhl požadované úrovně hlasitosti.

Veškeré změny provedené v nastavení zvuku se ukládají automaticky a nezávisle na aktuální úrovni hry. To znamená, že uživatelské preference hlasitosti budou zachovány i po ukončení a opětovném vstupu do hry.

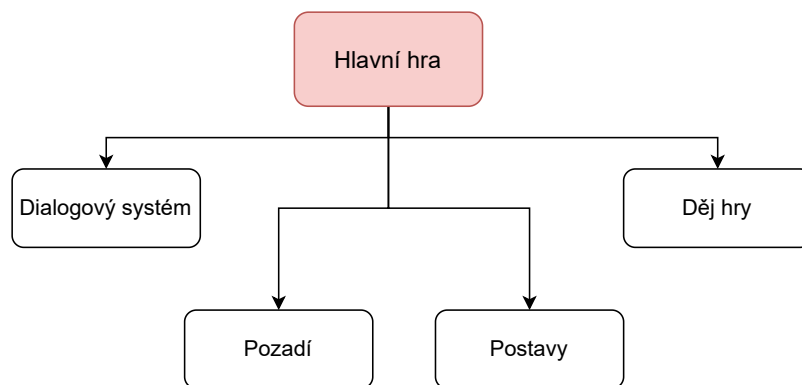


Obrázek 4.4: Na obrázku jsou zobrazeny tři vývojové diagramy znázorňující postupy nastavení programu nebo hry. První vývojový diagram ukazuje, jak scéna pracuje s nastavením a kontroluje, zda je nastavení uloženo v paměti. Pokud jsou nastavení uložena, přejdete k jejich použití; pokud ne, počkáte na nastavení a následuje jejich uložení, nebo pokud nebyla žádná nastavení zadána, použijete výchozí hodnoty. Druhý vývojový diagram znázorňuje použití nastavení ve hře; začíná podobnou kontrolou uložených nastavení a v závislosti na výsledku přechází k výchozím nastavením nebo k nastavením založeným na předchozích volbách uživatele. Vpravo dole je také drátový snímek okna nastavení se dvěma posuvníky pro nastavení zvuků a hudby.

Nastavení zvuku se aplikují okamžitě po jejich změně, což uživatelům umožňuje okamžitě posoudit dopad jejich změn na zvuk hry.

4.4 Hlavní hra

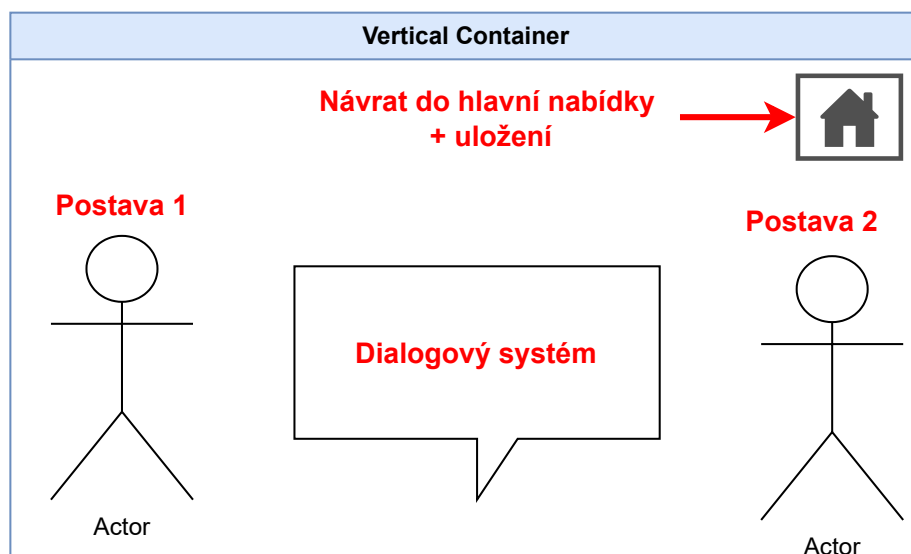
Hlavní část hry tvoří ilustrace a sprity ve stylu Digital Art a dialogový systém, kterým hráč prochází pomocí klikání myši. Na obrázku 4.5 jsou znázorněny všechny prvky, které tuto část tvoří.



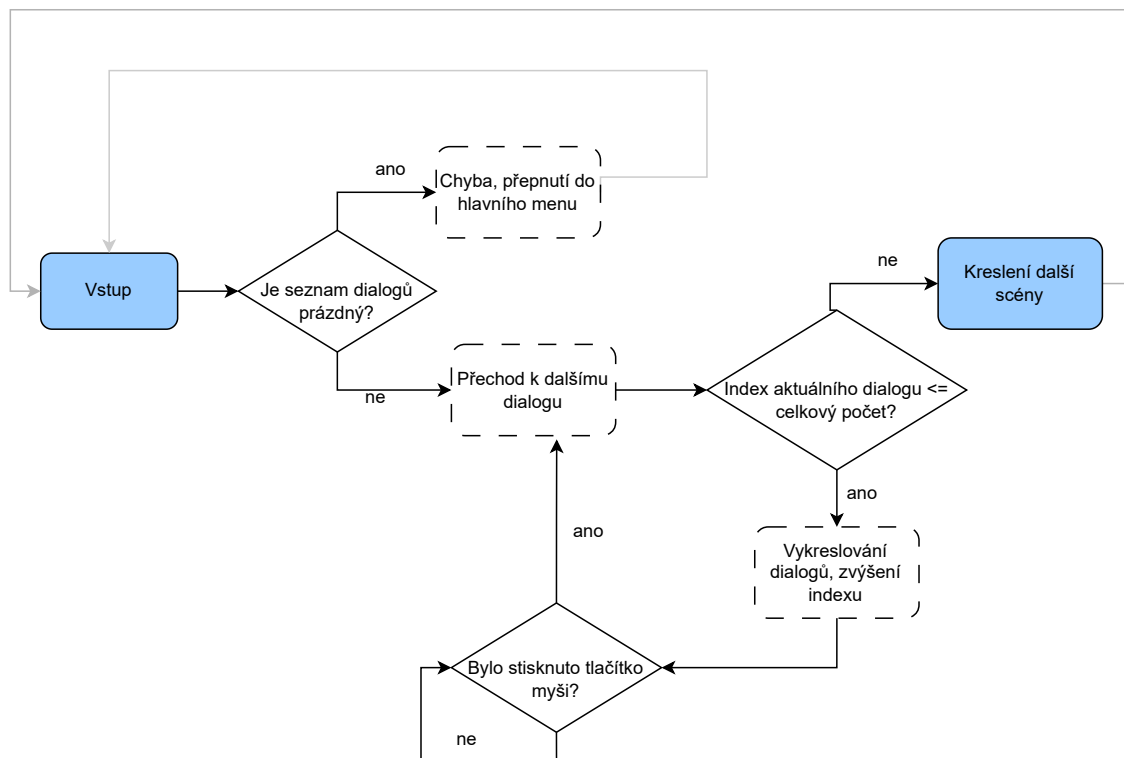
Obrázek 4.5: Hlavní hra se skládá z klíčových komponent, včetně *Dialogového systému*, který zajišťuje interaktivitu a prohlubuje příběh prostřednictvím dialogů s postavami. *Pozadí* a *Děj hry* určují vizuální a příběhovou atmosféru a umožňují hráči ponořit se do světa hry.

4.4.1 Dialogový systém

Dialogový systém představuje klíčový aspekt základních herních mechanismů, který hráči poskytuje možnost komunikovat s postavami a utvářet scénář. Pomocí uživatelského rozhraní zobrazeného na obrázku 4.6 a aktivace dialogů pomocí kliknutí myši může hráč plynule sledovat množství rozhovorů, činit rozhodnutí a přispívat k vývoji příběhu. Grafické znázornění logiky dialogového systému uvádí schéma 4.7.



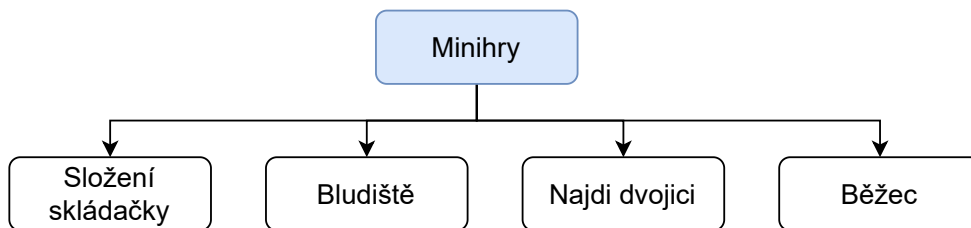
Obrázek 4.6: Rozhraní zahrnuje dva aktéry, *Postava 1* a *Postava 2*, kteří představují účastníky dialogových interakcí v rámci *Dialogového systému*, zajišťující dynamickou komunikaci a rozvoj příběhu. Komponenta *Návrat do hlavní nabídky + uložení* umožňuje uživatelům návrat do hlavní nabídky a uložení postupu. Tyto prvky společně vytvářejí funkční a intuitivní uživatelské rozhraní.



Obrázek 4.7: Blokové schéma popisuje algoritmus správy dialogů v aplikaci. Po přihlášení se zkontroluje, zda je seznam dialogů prázdný, a pokud je prázdný, přejde se do hlavní nabídky. Pokud seznam dialogů obsahuje položky, algoritmus postupně přechází na další dialog, reaguje na akce uživatele, jako je kliknutí myši, a pokračuje v zobrazování dialogů, dokud neprojdou všechny.

4.5 Mini hry

Hlavní zábavou výsledné aplikace jsou mini hry. Mini hry by neměly být příliš obtížné a jejich pochopení by nemělo trvat příliš dlouho. Některé mini hry mají časový limit, ale jsou i takové, které mohou trvat neomezeně dlouho. Při výběru vhodných mini her pro hraní byl brán zřetel na procvičování různých dovedností, přiměřenou obtížnost a celkovou zábavnost. Důležitým faktorem bylo také to, jak mini hra zapadá do tématu hry. Na schématu 4.8 jsou znázorněny hlavní mini hry, které tvoří hru.



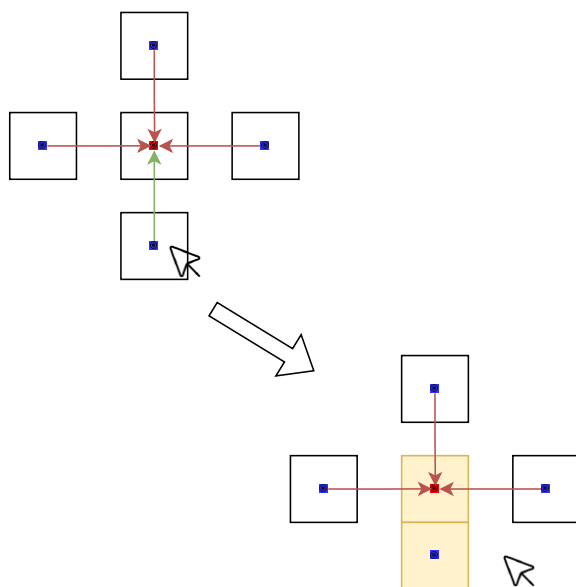
Obrázek 4.8: Schéma znázorňuje strukturu sekce *Minihry* jako hierarchické uspořádání, kde hlavním uzlem je kategorie *Minihry*, z níž se odvíjejí čtyři podkategorie her: *Bludiště*, *Najdi dvojici*, *Běžec* a *Složení mapy* a *Složení skládačky*. Každá podkategorie je uvedena jako samostatný blok s názvem hry, což zdůrazňuje jejich samostatnost v rámci celého systému.

4.5.1 Složení mapy

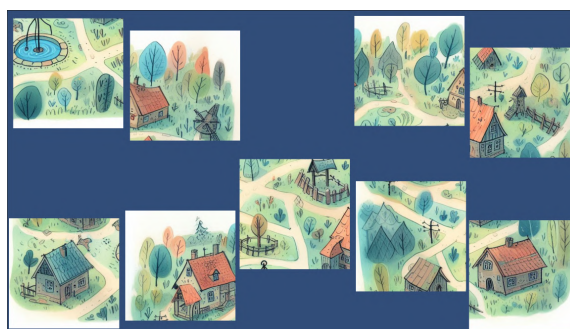
Minihra *Složení mapy* je znázorněna na obrázku 4.11 a skládá se z 9 čtvercových dílků stejného obrázku (viz Obrázek 4.10) poskládaných po celé hrací ploše. Úkolem hráče je rozpoznat, kam má být každý dílek umístěn, aby se obrázek složil. Hráč přesouvá dílky skládačky pomocí myši. Hra není časově omezena. Úroveň je považována za dokončenou, když hráč správně sestaví matici 3×3 dílků skládačky (obrázků). Navržený průběh minihry je zobrazen na diagramu 4.9. Navržený průběh minihry je zobrazen na diagramu 4.12



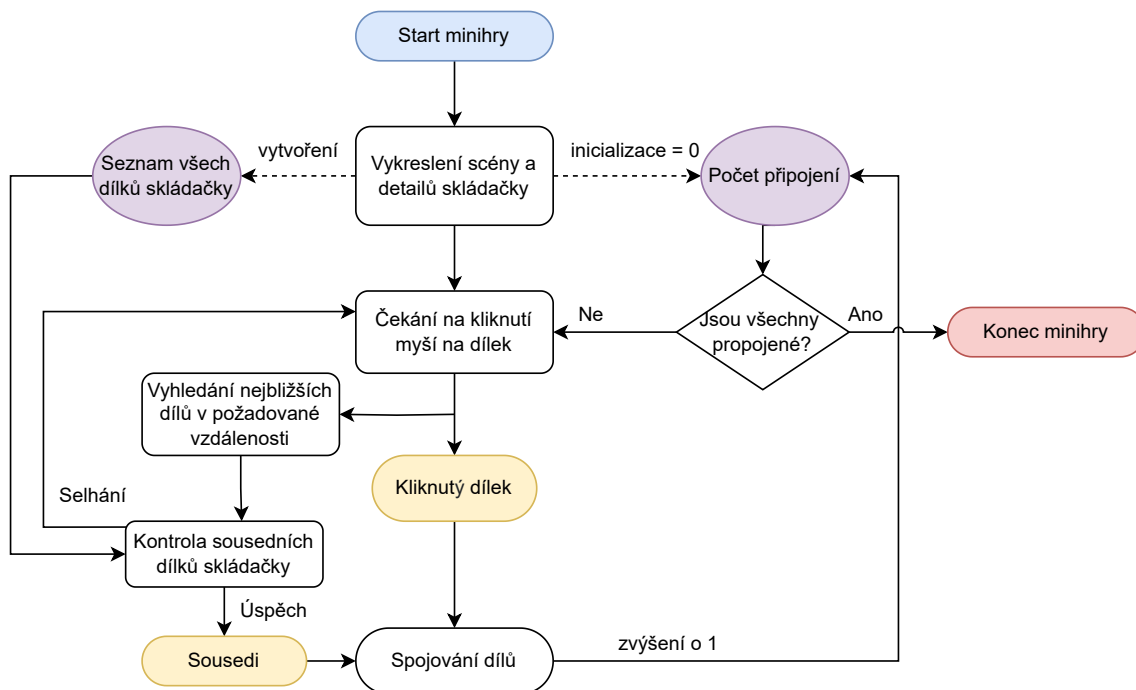
Obrázek 4.10: Obrázek správného rozmístění maticových karet 3×3 pro dokončení hry.



Obrázek 4.9: Schéma znázorňující logiku skládání puzzle. Spojení dvou dílků k sobě, když se k sobě přibližují.



Obrázek 4.11: Obrázek počátečního rozptýleného snímku karet skládačky na jevišti.

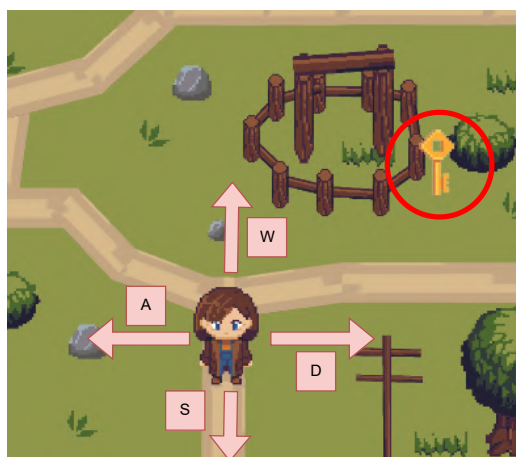


Obrázek 4.12: Toto schéma popisuje proces ovládání minihry ve videohře. Začíná inicializací minihry, po které následuje vykreslení scény a dílků skládačky. Uživatel se hry účastní klikáním myši na dílky skládačky, čímž se spustí proces hledání nejbližších spojitelných prvků a jejich následné propojení. Program kontroluje, zda jsou všechny dílky skládačky propojeny. Pokud je skládačka kompletně sestavena, minihra končí. Tento diagram vizualizuje logiku interakce mezi uživatelským rozhraním a hrou, jakož i zpracování akcí uživatele a posloupnost kroků k dokončení hry.

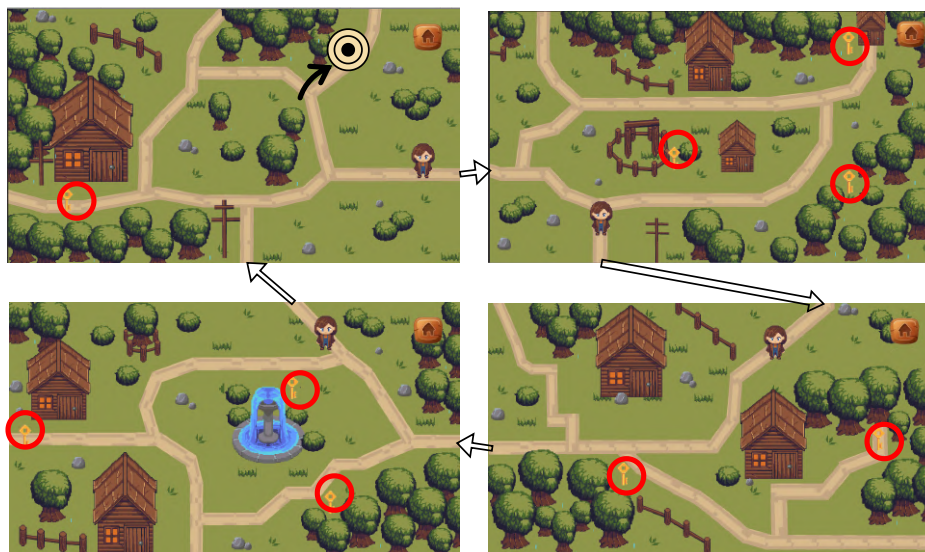
4.5.2 Otevřené bludiště

Minihra *Otevřené bludiště* nemá žádnou definitivní inspiraci. Hráč má pouze 30 sekund na to, aby si zapamatoval umístění klíčů na mapě a polohu kouzelné studny. Poté musí hráč posbírat všechny klíče a najít studnu. Postava se ovládá pomocí kláves W,S,A,D nebo šipek, což poskytuje pohodlné a intuitivní ovládání (viz Obrázek 4.13). Navržený průběh minihry je zobrazen na diagramu 4.15.

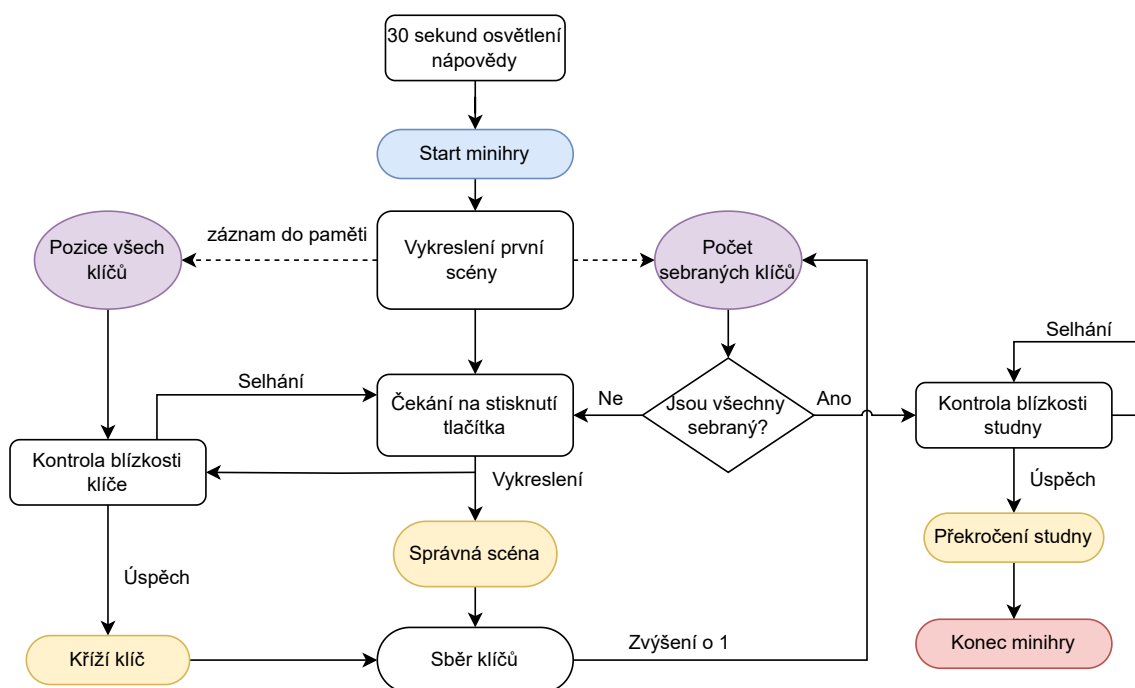
Celé bludiště je prezentováno jako velká mapa vesnice rozdělená do 4 scén. Každá scéna představuje čtvrtinu mapy a hráč, který se nachází na jedné ze scén, nevidí zbytek mapy. To vytváří pocit nepředvídatelnosti a výzvy, protože hráč neví, které cesty vedou do slepých uliček a které k cíli.



Obrázek 4.13: Příklad ovládání postavy pro pohyb po mapě a sbírání klíčů.



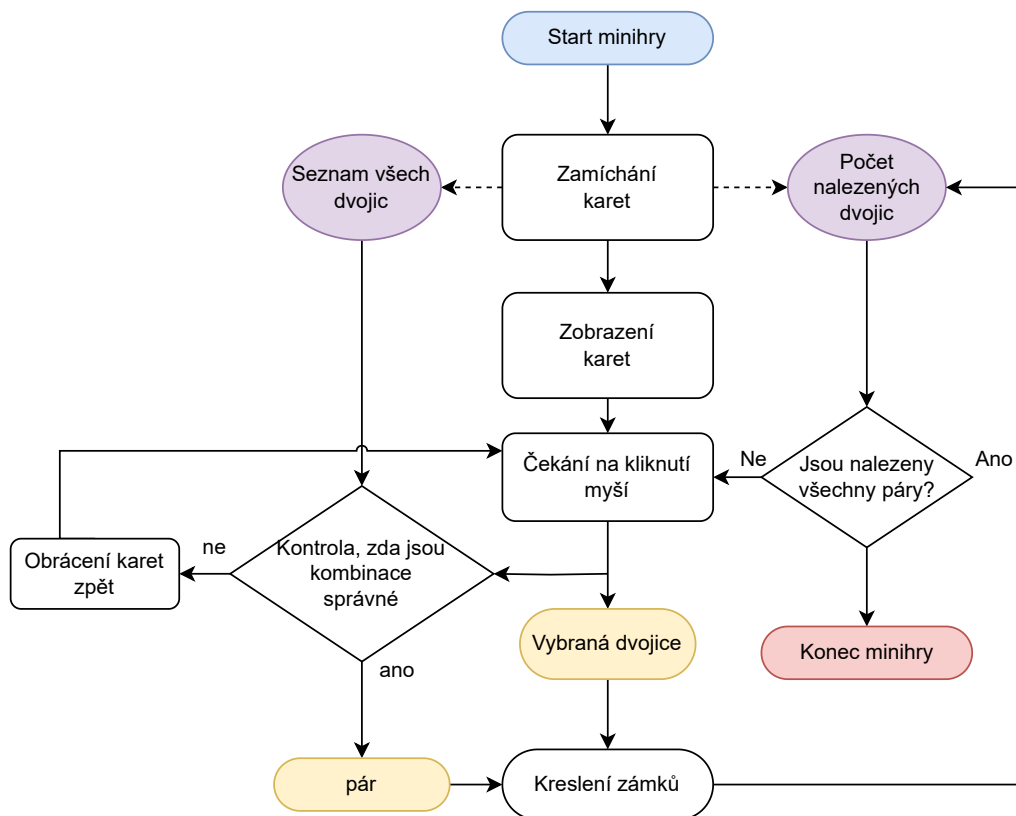
Obrázek 4.14: Rozložení mapy bludiště do 4 scén s přechodovými body mezi scénami.



Obrázek 4.15: Tento diagram popisuje proces ovládání minihry ve videohře. Začíná inicializací minihry, po níž následuje vykreslení scény. Uživatel se hry účastní stisknutím tlačítek na klávesnici, pohybem postavy po mapě, čímž se spustí proces hledání nejbližších kláves a jejich následné sbírání. Program kontroluje, zda byly všechny klíče sebrány. Pokud jsou všechny klíče sebrány, zkontroluje blízkost studny. Pokud je studna nalezena, minihra končí. Tento diagram znázorňuje logiku interakce mezi uživatelským rozhraním a hrou, jakož i zpracování akcí uživatele a posloupnost kroků k dokončení hry.

4.5.3 Najdi dvojici

Další fascinující minihra – digitální realizace oblíbené karetní hry Pexeso. Pravidla jsou jednoduchá: hráč otáčí karty, na jejichž zadní straně jsou ukryty obrázky klíčů. Úkolem hráče je najít dvojice stejných klíčů a otevřít všechny zámky. Návrh hry je znázorněn na obrázku 4.17.



Obrázek 4.16: Tento diagram popisuje logiku procesu pro hru v párování v rámci videohry. Hra začíná inicializací, která zahrnuje zamíchání karet. Po zamíchání karet hráč čeká na kliknutí myši, což vede k zobrazení karet. Jakmile hráč vybere dvojici karet, systém zkontroluje, zda jsou vybrané karty správné kombinace. Pokud kombinace není správná, karty se otočí zpět. Pokud kombinace odpovídá, počet nalezených dvojic se aktualizuje. Tento proces se opakuje, dokud nejsou nalezeny všechny páry karet. Jakmile jsou všechny páry nalezeny, minihra končí.

Klíčem ke hře je zapamatování si umístění klíčů na otočených kartách. V každém tahu musí hráč otevřít vždy dvě karty, a pokud objeví dvojici stejných klíčů, stanou se z nich zámky. Pokud se klíče liší, jsou otočeny zpět lícem dolů a hráč pokračuje ve svém tahu a snaží se zapamatovat si umístění klíčů na kartách.

Jedním ze zajímavých aspektů této hry je, že při každém novém zahájení hry se karty zamíchají v náhodném pořadí. Logika hry je znázorněna na obrázku 4.16



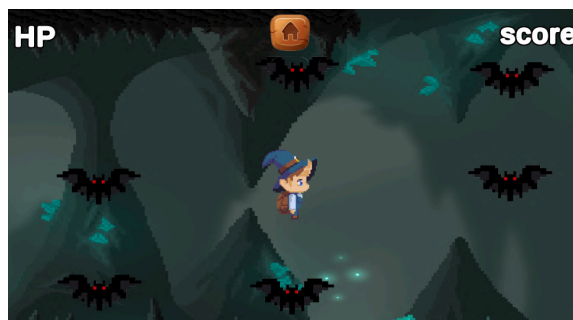
Obrázek 4.17: Příklad návrhu minihry. Vlevo je znázorněn začátek hry se všemi kartami lícem dolů a vpravo příklad některých nalezených párů.

4.5.4 Běžec

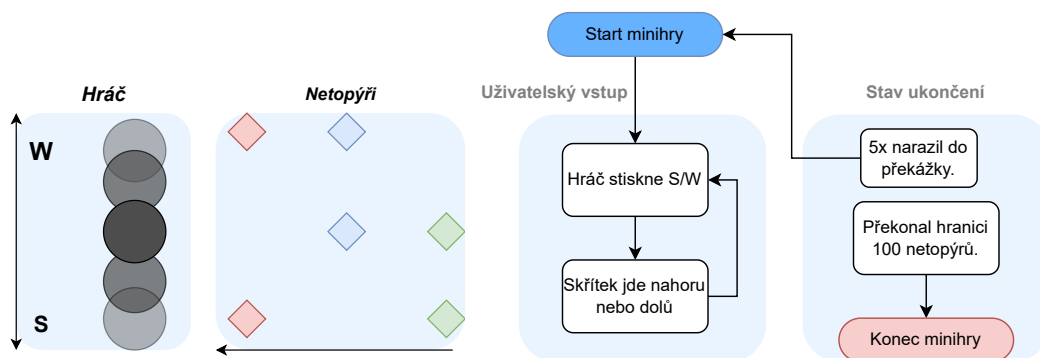
Jen málokdo neslyšel o kultovní klasické hře *Canabalt*^a. Verzi s kouzelným skřítkem Maxem představuje podzemní tunel, kterým musí projít tak, že se vyhýbá netopýrům, kteří ho blokují, jak lze vidět na návrhu 4.18.

Netopýři se objevují pravidelně a vždy jedním ze tří způsobů, takže skřítek může proletět/projít vzdálenost mezi nimi, jak je znázorněno na obrázku 4.19. Bez asistence Max poletí uprostřed, takže hráč musí gnóma ovládat pomocí kláves W, S a nastavit požadovaný směr.

^aOficiální stránka hry Canabalt – <https://canabalt.com/>

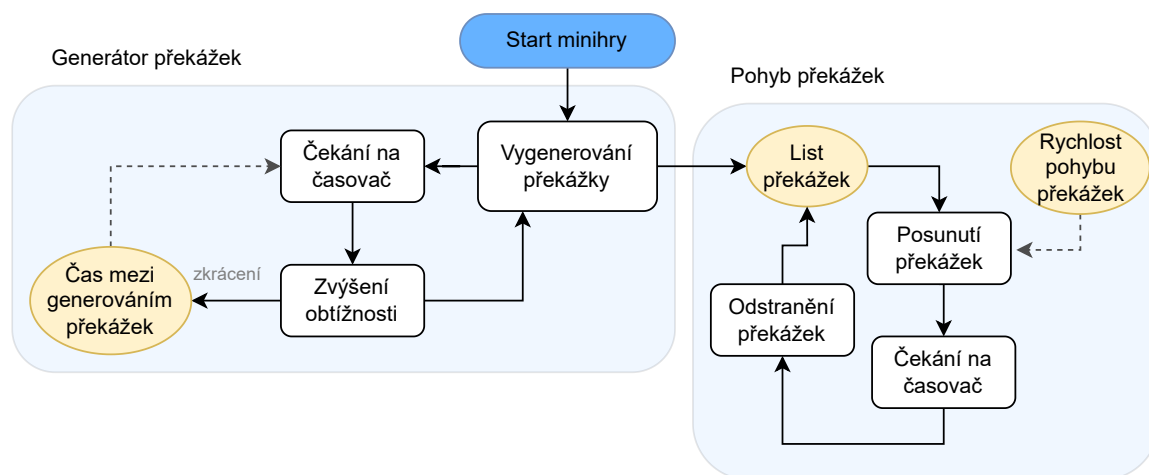


Obrázek 4.18: Vizualní koncept minihry Běžec.



Obrázek 4.19: V této hře se hráč zobrazený na levém obrázku pohybuje pouze nahoru a dolů, zatímco překážky se generují jedním ze tří způsobů a pohybují se směrem k hráči. Jediným vstupem uživatele je stisknutí klávesy W nebo S. Minihra končí úspěchem, jakmile hráč projde 100 netopýrů, a neúspěchem, jakmile na netopýra narazí pětkrát.

Netopýři se objevují pravidelně s různou výškou mezi nimi. Doba mezi jednotlivými objeveními se postupně zkracuje (s každou sekundou hry se doba mezi generacemi zkracuje o 0,5 sekundy). Celý průběh minihry lze vidět na diagramu 4.20.

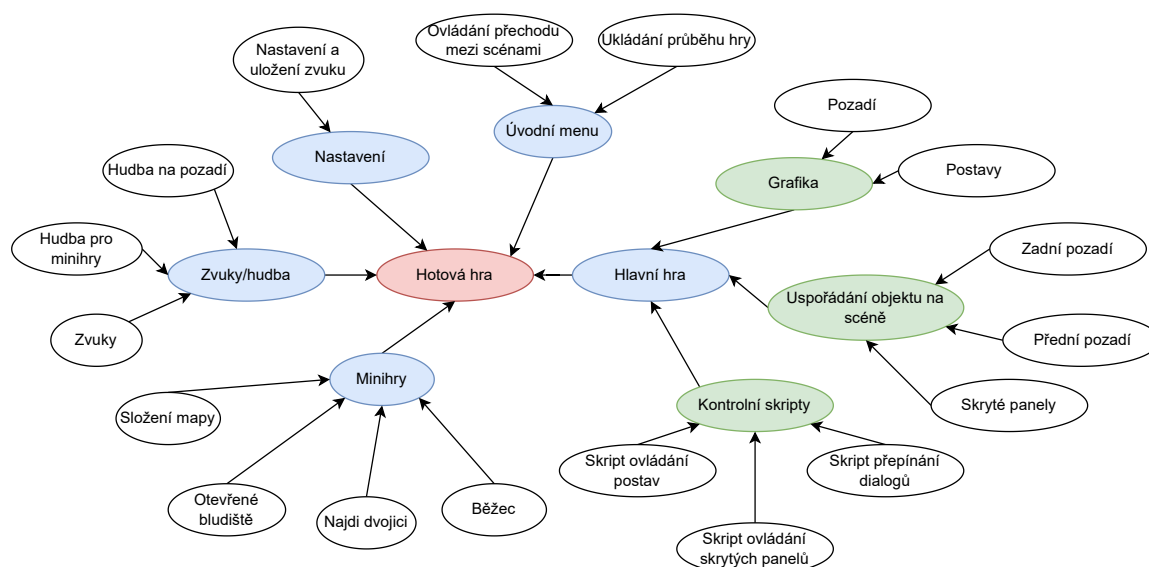


Obrázek 4.20: Tento schematický postup popisuje logiku minihry na příkladu běžecké hry, ve které se hráč musí vyhýbat překážkám. Hra začíná inicializací, která zahrnuje generování prvních překážek. Po jejich vygenerování hra vstoupí do čekacího cyklu na časovač, který reguluje čas mezi dalšími generacemi překážek. Postupem času se obtížnost zvyšuje tím, že se zkracuje doba mezi generováním překážek, čímž se hra stává náročnější a dynamičtější. Překážky se pohybují ve směru pohybu hráče. V případě kolize s překážkou lze překážky odstranit ze seznamu aktivních překážek na obrazovce. Překážky se také po určité době z obrazovky odstraní.

Kapitola 5

Realizace

Tato kapitola popisuje implementaci herních mechanismů uvedených v předchozích kapitolách. Bylo provedeno mnoho samostatné práce, která byla následně spojena dohromady. Začalo to nakreslením všech spritů a obrázků a skončilo skriptováním, dialogy a sestavením hotové hry. Průběh vývoje je znázorněn na Obrázku 5.1. A průběh vývoje miniher je uveden dále v části 5.2.



Obrázek 5.1: Diagram znázorňující aktuální průběh vývoje aplikace. Nejprve byla implementována grafika a hlavní hra. To však vedlo k podcenění množství práce potřebné k realizaci miniher. Grafika byla vytvářena v průběhu celého procesu vývoje a v druhé polovině vývoje se začalo výrazně zapojovat skriptování. Kvůli zdlouhavému studiu procesu tvorby hry zbývalo příliš málo času na celkové testování.

Jako vývojové prostředí pro implementaci aplikace bylo zvoleno herní vývojové prostředí *Unity Engine*¹. Hlavním důvodem pro výběr tohoto prostředí je velikost vývojářské komunity, dostupnost výukových programů a také podrobnější popis v části 3.3. K vytvoření

¹Odkaz na stránky vývojářského prostředí Unity Engine – <https://unity.com>

spritu ve stylu 2D pixel art bylo použito prostředí *Aseprite*². Ostatní grafické prvky, jako je grafické uživatelské rozhraní nebo postavy, byly vytvořeny v prostředí *Paint tool Sai*³.

5.1 Hlavní hra

Při realizaci hlavní hry bylo potřeba implementovat prvky popsané v návrhu 4.5. Nejprve byly vygenerovány všechny postavy pomocí umělé inteligence, konkrétně pomocí online nástroje *Bing Image Creator*⁴, Obrázek 5.2. A poté byly překresleny pomocí čtyř hlavních potřebných emocí: normální stav, strach (překvapení), hněv, smutek – Obrázek 5.3.



Obrázek 5.2: Postavy generované umělou inteligencí pro inspiraci.

²Odkaz na stránku aplikace *Aseprite* – <https://www.aseprite.org/>

³Odkaz na stránku aplikace *Paint tool Sai* – <https://www.systemax.jp/en/sai/>

⁴Odkaz na online nástroje *Bing Image Creator* – <https://www.bing.com/images/create>



Obrázek 5.3: Postavy nakreslené nástrojem *Paint tool Sai* s různými emocemi.

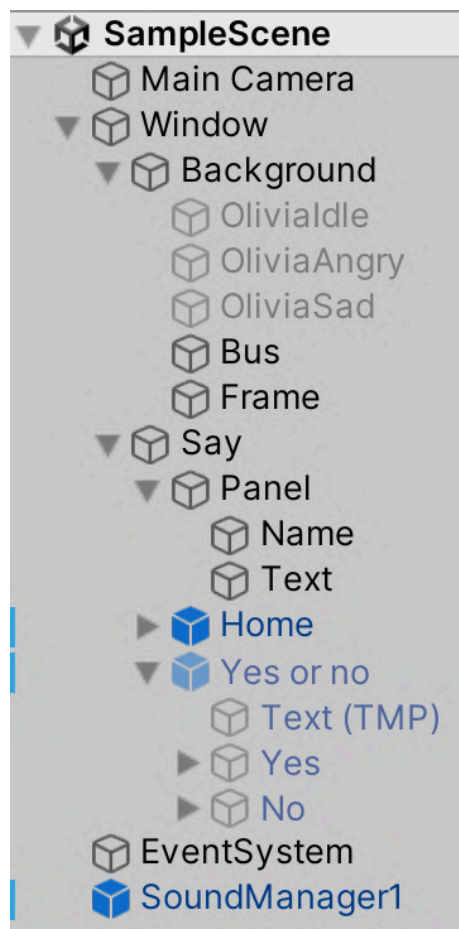
5.1.1 Uspořádání objektu ve scéně

Všechny scény hlavní hry jsou uspořádány stejně, jedna z nich je uvedena jako příklad (viz Obrázek 5.4). V této herní scéně, nazvané *SampleScene*, jsou objekty uspořádány do kategorií, které odrážejí strukturu scény a její interaktivní prvky. Hlavními skupinami jsou *Background*, *Say* a *Home*. *Background* obsahuje prvky pozadí, jako je různé emocionální stavy postavy *Olivia* – *OliviaIdle*, *OliviaAngry*, *OliviaSad*. Dále obsahuje obrázky, jako je *Bus* a *Frame*. Tyto objekty se stávají aktivními v závislosti na průběhu hry. *Say* ovládá systém dialogů, včetně *Panel* se jménem a textem, který se zobrazuje během komunikace ve hře.

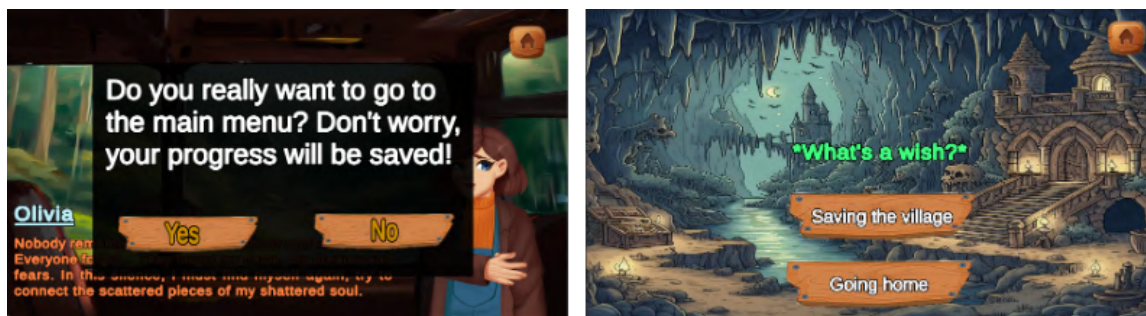
Zpočátku jsou aktivní pouze *Background* a dialogový systém *Say*. Postavy na scéně jsou v neaktivním stavu a aktivují se pouze tehdy, když je potřeba jejich účast v dialogu, poté jsou opět neaktivní. To umožňuje dynamickou interakci mezi postavami a hráčem.

Výběrem možnosti *Home*, která hráče může vrátit do hlavní nabídky, se aktivuje skrytý prvek *Text (TMP)* v rámci *Yes or No*, který je součástí systému potvrzení volby hráče (viz Obrázek 5.5). Tato akce zneaktivní dialogový panel a místo něj poskytne možnosti přechodu do hlavní nabídky.

Po dokončení všech dialogů a interakcí hra přejde k další scéně. *EventSystem* slouží k ovládní kliknutí myši na scéně a *SoundManager1* slouží k ovládní zvukové stopy scény, plynulému přechodu hudby na pozadí a zapnutí zvukových efektů v případě potřeby.



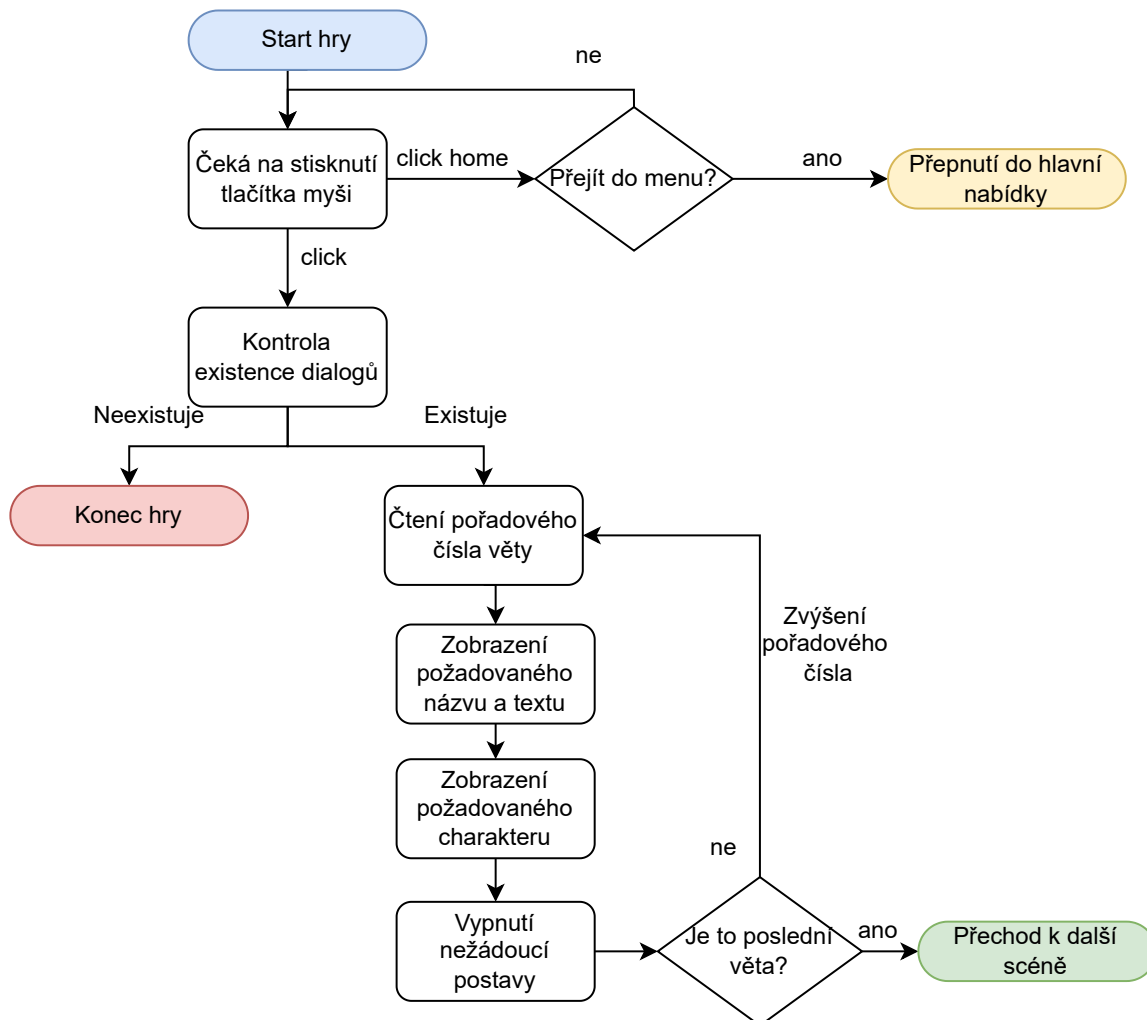
Obrázek 5.4: Příklad umístění objektů na hlavní herní scéně. Vizualizace grafiky a dialogového systému.



Obrázek 5.5: Příklad aktivace a deaktivace skrytých panelů. Obrázek vlevo ukazuje aktivaci panelu pro objasnění přechodu do hlavní nabídky. Obrázek vpravo ukazuje deaktivaci panelu s dialogovým systémem a aktivaci panelu výběru.

5.1.2 Kontrolní skripty

Klíčovou roli hraje ovládání hlavních herních procesů pomocí skriptů, které je realizováno relativně jednoduše. Logika řídicích skriptů je uvedena v diagramu 5.6. Schéma znázorňuje logiku hlavních herních operací, která je vtělena do těchto řídicích skriptů.

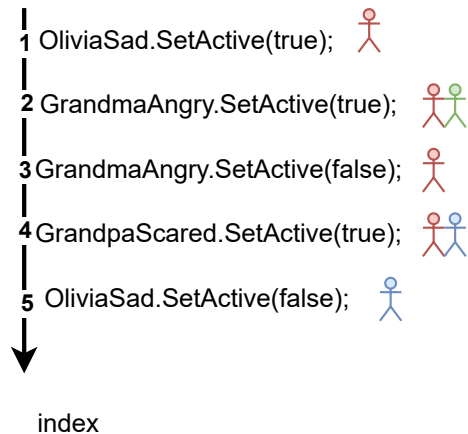


Obrázek 5.6: Tento diagramový tok popisuje logiku řízení scénáře v interaktivní videohře. Hra začíná inicializací, která zahrnuje kontrolu přítomnosti dialogů. Pokud jsou dialogy přítomny, systém přečte pořadové číslo věty a zobrazí na obrazovce požadovaný název a text a odpovídající znak. Pokud není přítomen žádný dialog, kontrola přejde k dokončení hry. Pakliže se nejedná o poslední frázi, systém zvýší pořadové číslo a přejde k další scéně. Pokud se jedná o poslední větu, hra čeká na stisknutí tlačítka myši. Po stisknutí tlačítka *Home* se aktivují skryté panely a dojde k přechodu do hlavní nabídky, čímž se uzavře aktuální scéna.

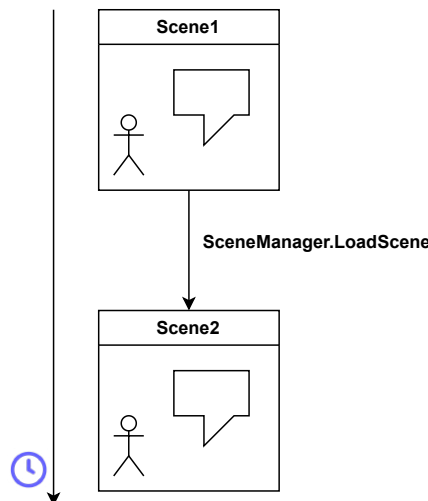
V Unity se k vytvoření interaktivního dialogového systému používá několik klíčových komponent a funkcí, které řídí průběh hry a interakci s uživatelem.

Metoda *SetActive* slouží k zapnutí nebo vypnutí objektů na scéně (viz Obrázek 5.7). V Unity je každý objekt na scéně instancí třídy *GameObject*. Metoda *SetActive* mění stav aktivity objektu: je-li nastavena na hodnotu *true*, objekt se stane aktivním (viditelným a funkčním ve hře), je-li nastavena na hodnotu *false*, stane se neaktivním (není viditelný a neovlivňuje hru). Ve skriptu se tato metoda používá k řízení zobrazení postav podle toho, kdo se právě účastní dialogu.

SceneManager je třída, která v Unity spravuje scény (viz Obrázek 3.3). Metoda *LoadScene* načte novou scénu na základě jejího názvu nebo indexu. Ve hrách se scény často používají k vymezení úrovní, nabídek a dalších částí hry. Ve skriptu se metoda *SceneManager.LoadScene* používá k načtení další scény (viz Obrázek 5.8).



Obrázek 5.7: Příklad použití metody *SetActive* v projektu.



Obrázek 5.8: Příklad použití metody *SceneManager.LoadScene* v projektu.

5.2 Minihry

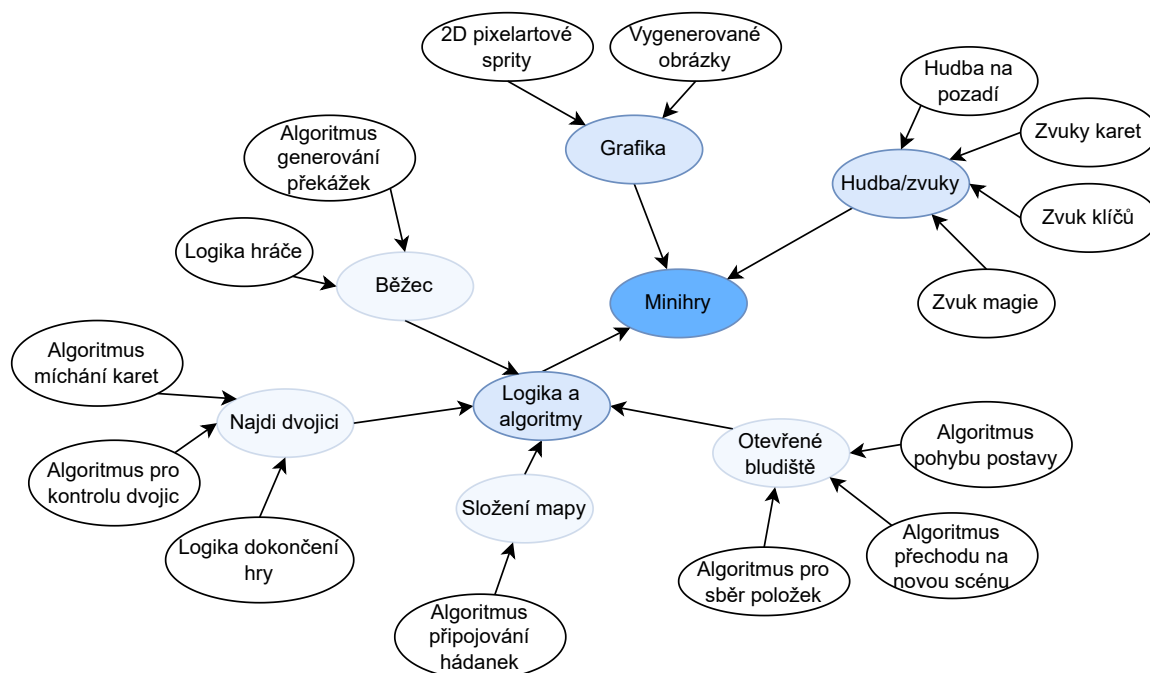
Každá minihra byla vyvíjena samostatně a postupně. Pro každou z nich byly speciálně vytvořeny sprity ve stylu 2D pixelové grafiky podle postav (viz Obrázek 5.9). Pixel art je způsob digitálního umění, ve kterém jsou detaily obrázku s vysokým rozlišením redukovány pomocí malého počtu pixelů, čímž dochází k jejich abstrakci [10]. Kromě toho byly použity technologie umělé inteligence k vygenerování dalších potřebných prvků, jako jsou mapy, klíče, karty zápalek, zámky a další. Konkrétně byl využit online nástroj Bing Image Creator⁵.

⁵Odkaz na online nástroj *Bing Image Creator* – <https://www.bing.com/images/create>

Po dokončení všech prvků byl zahájen vývoj základních algoritmů pro minihry a psaní příslušných skriptů. V této části jsou analyzovány klíčové algoritmy použité v jednotlivých minihrách. Všechny vyvinuté prvky pro minihry jsou zobrazeny na Obrázku 5.10. Hra obsahuje 4 minihry: Skládačka, Otevřené bludiště, Najdi dvojici (Pexeso) a Bězec. Každá z nich je podrobněji popsána níže.



Obrázek 5.9: Sada spritů postav navržených pro minihry ve stylu 2D pixelové grafiky. Každý sprite představuje různé pózy postav a animační stavy používané ve hře.

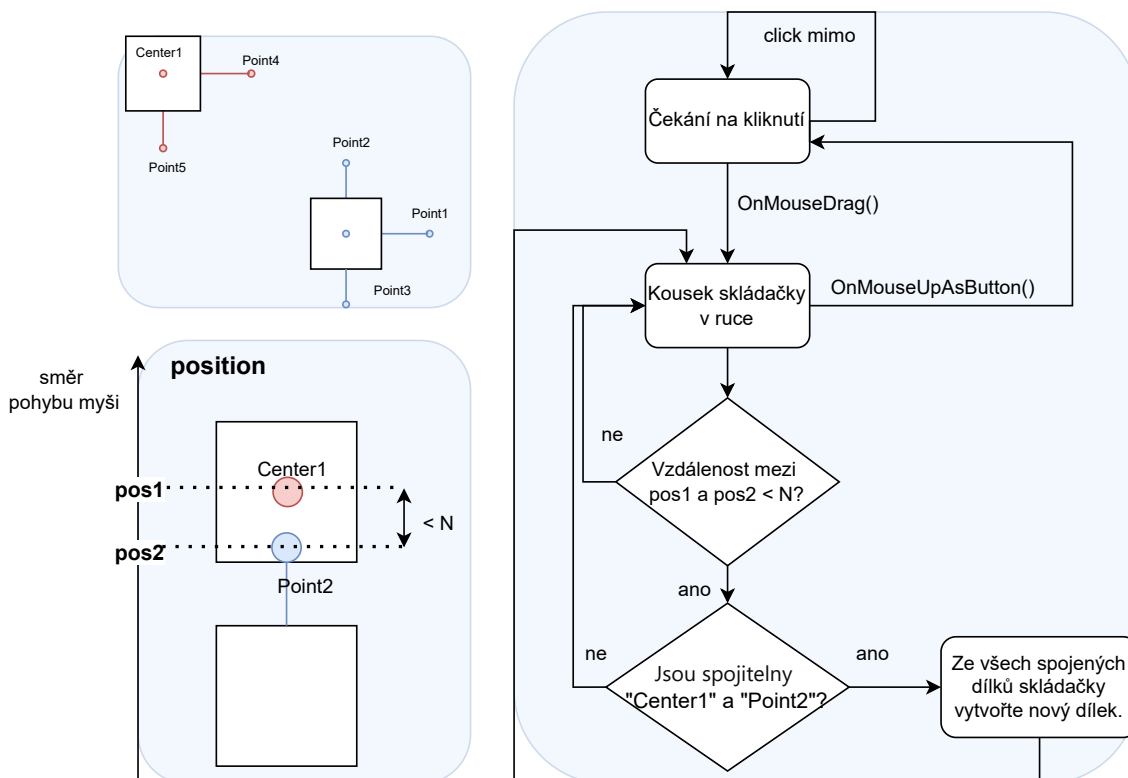


Obrázek 5.10: Schéma ukazuje strukturu tvorby minihry, včetně algoritmů pro herní logiku, generování předmětů a správu postav. Algoritmy zahrnují míchání karet, kontrolu dvojic, sbírání předmětů a přechod mezi scénami. Grafická složka je realizována prostřednictvím 2D pixelové grafiky, včetně spritů a generovaných obrázků, zatímco zvuk je navržen prostřednictvím hudby na pozadí a různých zvukových efektů.

5.2.1 Složení mapy

V rámci minihry je hlavním úkolem sestavit puzzle, které je reprezentováno samostatnými prvky (dílků), z nichž každý obsahuje sprite. Podstatou hry je spojování těchto dílků pomocí myši, která umožňuje pohybovat prvky po hrací ploše.

Každý dílek skládačky je vybaven několika spojovacími body (jedním středovým a dvěma až čtyřmi bočními, podle typu skládačky), které je třeba spojit s odpovídajícími body na ostatních dílcích skládačky. Během hry, když uživatel táhne za dílek skládačky, může jej spojit s jiným, pokud jsou spojovací body ve vzájemné blízkosti a splňují určité podmínky spojení (viz Obrázek 5.11).



Obrázek 5.11: Diagram znázorňuje proces spojování dílků skládačky v herní mechanice. Začíná od okamžiku, kdy hráč uchopí dílek skládačky. Pokud hráč klikne myší mimo objekt, akce se přeruší. Během procesu přetahování (*OnMouseDrag*) hráč posouvá dílek skládačky a systém průběžně kontroluje, zda je druhý v přijatelné vzdálenosti menší než N . Když jsou dva prvky skládačky dostatečně blízko sebe, systém kontroluje, zda body spojení odpovídají předem definovanému pravidlu. Pokud se podmínky shodují, prvky skládačky se spojí a vytvoří novou jedinou strukturu. Tento proces se opakuje, dokud nejsou všechny odpovídající prvky spojeny do jednoho kompletního objektu.

Softwarová logika určí, zda lze dva body spojit, na základě jejich názvů (původně nastavených ručně) a polohy. Pokud jsou podmínky spojení splněny, skládačky se spojí a úspěšné spojení potvrdí zvukový efekt. Po úspěšném spojení hlavolamů v Unity se vytvoří nový nadřazený objekt, který se stane kontejnerem pro spojené hlavolamy. Tím se změní hierarchie objektů a zajistí se, že celý komplex je spravován jako jedna entita.

Jakmile jsou všechny dílky spojeny, aktivuje se vítězný panel a po krátké prodlevě se hráč přesune na další scénu, což symbolizuje splnění úkolu. Podrobně je logika hry a algoritmus znázorněn na obrázku

Tento algoritmus používá dvě metody Unity *OnMouseDown()* a *OnMouseDownAsButton()*, které se obvykle používají k řízení interakce uživatele s objekty pomocí myši. Tyto metody jsou zabudovány do systému obsluhy událostí a jsou určeny k implementaci interaktivních funkcí, jako je přetahování myši a obsluha kliknutí.

Metoda *OnMouseDown()* se aktivuje během každého snímku, když uživatel stiskne a podrží tlačítko myši a přesune kurzor, přičemž zůstane nad objektem s aktivovanou komponentou *Collider* [7, str. 54]. V tomto algoritmu se tato metoda používá k implementaci funkce přetahování objektů skládačky.

Metoda *OnMouseDownAsButton()* je zavolána, když uživatel uvolní tlačítko myši po kliknutí na objekt s komponentou *Collider* [7, str. 54], pokud kurzor nebyl přesunut mimo hranice objektu. V algoritmu se tato metoda používá k signalizaci dokončení procesu přetahování skládačky. To umožňuje systému určit, že skládačka již není v aktivním pohybu a je připravena na nové interakce nebo následné akce uživatele.

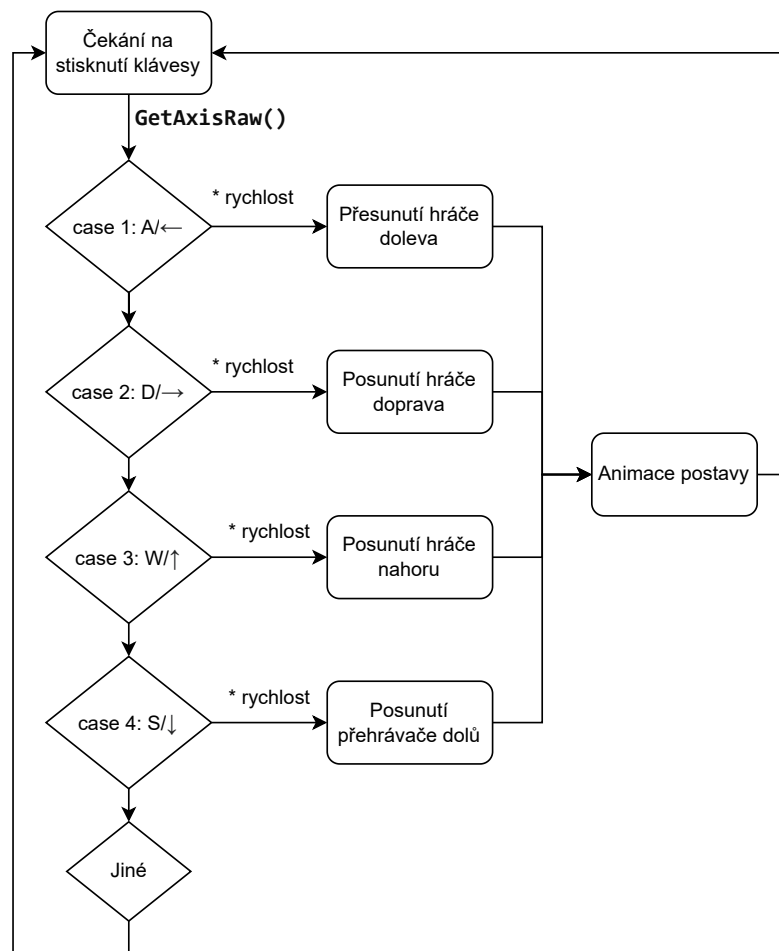
5.2.2 Otevřené bludiště

V této minihře jsou tři hlavní algoritmy, na kterých je založen herní prostor: pohyb postavy, pohyb mezi částmi mapy (scénami) a sbírání předmětů. Každý z těchto algoritmů je podrobněji popsán níže.

Algoritmus pohybu postavy

Tento algoritmus využívá kombinaci uživatelského vstupu, animace a fyzikální mechaniky k ovládní pohybu postavy v Unity. Nejprve se inicializuje komponenta odpovědná za zpracování fyzikálních vlastností objektu, jako je pohyb a kolize. Horizontální a vertikální pohyb je řízen na základě uživatelského vstupu, který je průběžně zpracováván, což umožňuje tvarovat směr pohybu. Tyto informace se také používají k přizpůsobení animací, takže pohyb postavy je vizuálně plynulý a přirozený.

Vlastní pohyb postavy se provádí v rámci zpracování fyziky, kdy se na základě směru a rychlosti vypočítá nová poloha. Tento proces zajišťuje stabilní a přesný pohyb, který odpovídá fyzikálním zákonům herního světa. Logika algoritmu je znázorněna v diagramu 5.12.



Obrázek 5.12: Schéma popisuje ovládání pohybu postavy pomocí funkce *GetAxisRaw()*, která reaguje na stisky kláves (A/←, D/→, W/↑, S/↓). V závislosti na zvoleném směru pohybu se postava pohybuje v příslušném směru a animace se aktivuje s ohledem na nastavenou rychlost.

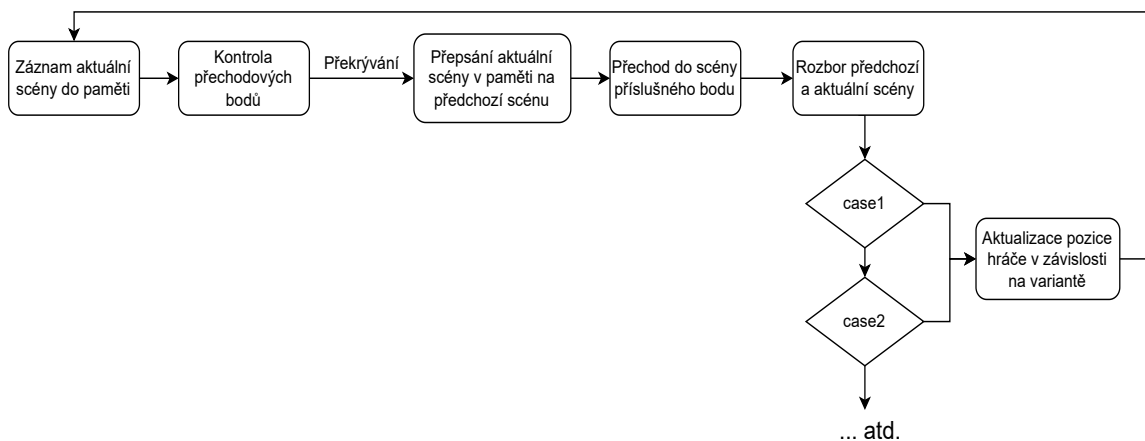
V Unity se funkce *Input.GetAxisRaw* [13, str. 268] používá k přímému příjmu vstupních dat od uživatele a vrací hodnoty od -1 do 1 bez dalšího zpracování. Pro vstup z klávesnice budou tyto hodnoty -1, 0 nebo 1 v závislosti na stisknutých klávesách. Pro vstup z klávesnice je osa *Horizontal* obvykle spojena s klávesami A a D, kde A nastavuje hodnotu -1 a D nastavuje hodnotu 1. To zajišťuje horizontální pohyb doleva, resp. doprava. Svislá osa *Vertical* je obvykle spojena s klávesami W a S, kde W nastavuje hodnotu 1 a S nastavuje hodnotu -1. Svislá osa je obvykle spojena s klávesami W a S. Tím je zajištěn vertikální pohyb vpřed, resp. vzad. To umožňuje náhlé změny hodnot bez jakéhokoli meziprojektu, což může být užitečné při implementaci vlastní logiky zpracování vstupu z klávesnice. V bludišti tato funkce řídí pohyb postavy načtením horizontálních a vertikálních vstupů pro určení směru pohybu. Tato data jsou předána animátoru, který mění animace na základě směru a rychlosti pohybu. Funkce *FixedUpdate* [7, str. 73] slouží k aktualizaci fyzické polohy postavy na základě její rychlosti a času, čímž zajišťuje plynulý pohyb.

Algoritmus přechodu na novou scénu

Vyvinutý algoritmus přechodu scén založený na Unity se zaměřuje na zajištění kontinuity a kontextové integrity hraní při pohybu postavy mezi různými částmi mapy. Pomocí třídy *SceneManager* algoritmus extrahuje název aktivní scény a mapuje jej na uloženou hodnotu předchozí scény uložené v *PlayerPrefs* (popsáno níže). To umožňuje systému určit kontext, ze kterého uživatel vstupuje do nové lokace.

Hlavní tělo metody *Start* obsahuje podmíněné příkazy, které kontrolují, zda se aktuální a předchozí scéna shodují. V závislosti na této shodě se upraví poloha postavy tak, aby její pohyb po mapě logicky a prostorově pokračoval. Pokud se například postava přesune ze scény vlevo do scény vpravo, její počáteční pozice v nové scéně bude nastavena vlevo, což by mělo odpovídat její koncové pozici v předchozí scéně.

Po nastavení počáteční pozice se název aktuální scény uloží do *PlayerPrefs* pod klíč *PreviousScene*. Tato hodnota bude při příštím spuštění scény použita k opětovnému přizpůsobení pozice postavy podle jejího posledního umístění v předchozí scéně, čímž bude zachována celistvost a plynulost hraní. Logika algoritmu je znázorněna v diagramu 5.13.



Obrázek 5.13: Schéma popisuje proces správy přechodů mezi scénami v herní aplikaci. Proces začíná zápisem aktuální scény do paměti, což umožňuje systému sledovat změny a spravovat body přechodu. Algoritmus analyzuje aktuální a předchozí scénu, aby určil potřebné úpravy umístění postavy. Poté systém v závislosti na zvoleném scénáři aktualizuje polohu postavy a přejde na novou scénu. Nakonec se informace o aktuální scéně aktualizují v paměti jako o scéně předchozí, čímž je zajištěna aktuálnost údajů pro následující přechody. Tento přístup zaručuje plynulé a konzistentní hraní.

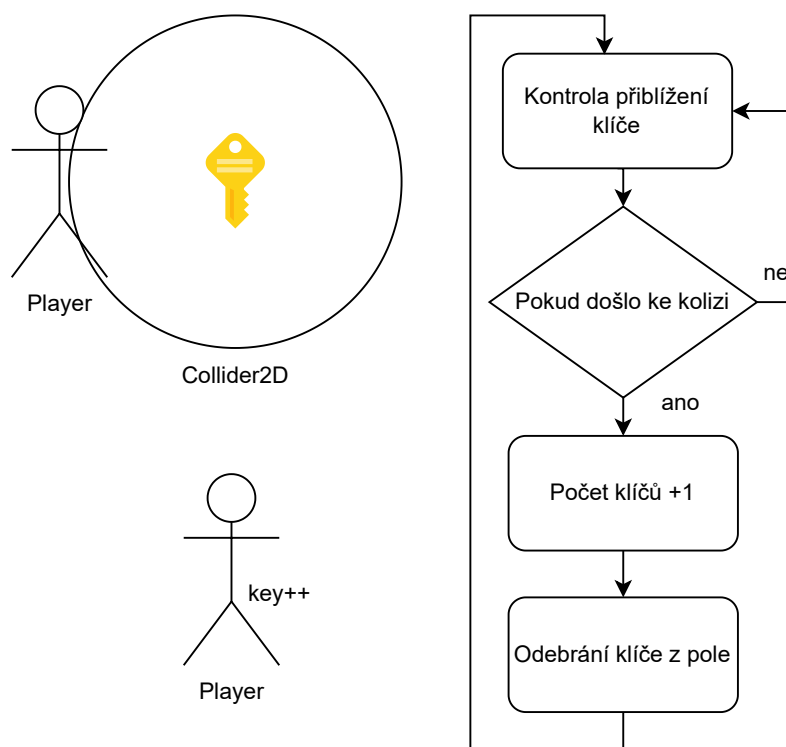
PlayerPrefs v Unity je systém ukládání hodnot klíčů, který slouží k ukládání uživatelských nastavení a předvoleb mezi herními relacemi. V tomto algoritmu se *PlayerPrefs* používá k uložení názvu poslední scény, což pomáhá určit výchozí pozici postavy při přechodu mezi scénami. Ukládání a načítání dat, například herní úroveň nebo nastavení, se provádí pomocí příkazů jako *PlayerPrefs.SetInt* a *PlayerPrefs.GetInt*.

Algoritmus pro sběr předmětů

Tento algoritmus spravuje mechanismus shromažďování klíčů ve hře, včetně ukládání a načítání stavů mezi herními relacemi. Při inicializaci objektu se nastaví spouštěč pro jeho

kolizní složku. Pokud byl klíč již dříve shromážděn, což je indikováno uloženou hodnotou 1 pro tento klíč, je objekt zničen, aby se zabránilo jeho opětovnému sebrání.

Pokud hráč aktivuje spouštěč interakcí s objektem a pokud je tento objekt označen jako hráč, zvýší se počet klíčů. Změny jsou zaznamenány v úložném systému, je spuštěn zvukový soubor a objekt je odstraněn z hracího pole. Tím je zajištěno logické pokračování herního procesu bez možnosti opakovaného sběru stejného klíče. Logika algoritmu je znázorněna v diagramu 5.14



Obrázek 5.14: Schéma popisuje zpracování události sběru klíčů. Událost se spustí, jakmile je prostřednictvím komponenty Collider2D detekována kolize, která naznačuje, že hráč může interagovat s objektem klíče. Jakmile je kolize detekována, provede se kontrola, zda se jedná o hráče (Tag = *Player* [7, str. 57]), aby se vyloučila interakce s necílovými objekty (například s geometrií scény). Pokud je účast hráče potvrzena, zvýší se počet klíčů, které hráč nasbíral, o jeden. Posledním krokem je odstranění klíčového objektu z herní arény, čímž se zabrání jeho dalšímu sběru.

Collider2D [7, str. 148] je fyzikální komponenta, která řídí tvar srážek 2D objektů. Lze ji nakonfigurovat tak, aby fungovala jako spouštěč a nevyvolávala fyzikální reakce při srážkách, ale přesto oznamovala průsečíky prostřednictvím událostí.

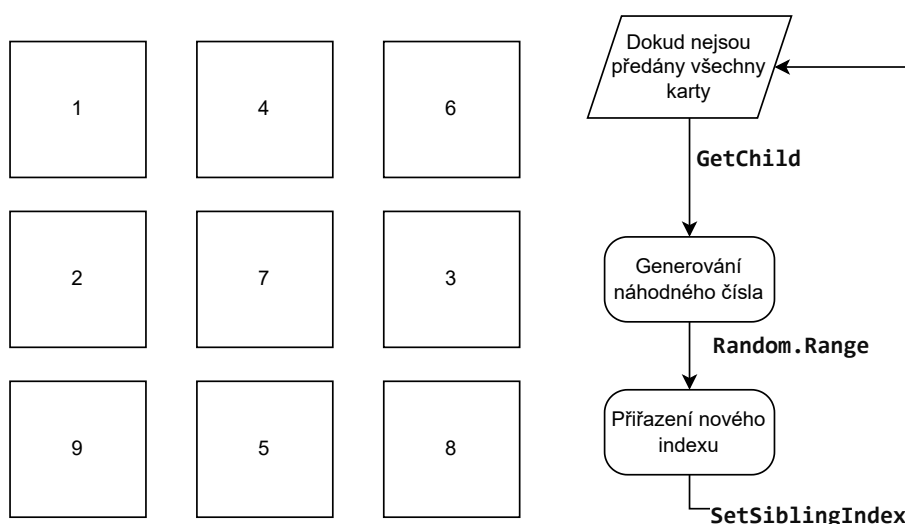
OnTriggerEnter2D [7, str. 154, 295] je metoda, která se automaticky zavolá, když do oblasti tohoto spouštěče vstoupí jiný objekt s komponentou Collider2D v režimu spouštěče. V kontextu algoritmu, když hráč (který musí mít značku *Player* a komponentu *Collider2D* nakonfigurovanou jako spouštěč) vstoupí do klíčového kolideru, aktivuje se metoda *OnTriggerEnter2D*. Tato metoda implementuje logiku, která zvýší počítadlo klíčů o jedna, uloží tuto změnu do *PlayerPrefs* pro budoucí použití ve hře, přehraje zvukový efekt a zničí objekt klíče ve hře, aby se zabránilo jeho dalšímu shromažďování.

5.2.3 Najdi dvojici (Pexeso)

Implementace minihry Najdi dvojici je založena na třech základních logických algoritmech: algoritmu pro míchání karet, algoritmu pro kontrolu správných dvojic a logice pro ukončení hry. Každý z nich je podrobněji popsán níže.

Algoritmus míchání karet

Metodika používaná k promíchání karet spočívá ve změně pořadí podřízených objektů panelu karet. Při inicializaci objektu karty se systematicky prohledávají všechny umístěné na nadřazeném panelu. Pro každou kartu je v hierarchii panelu nastaven nový index, který je určen náhodně. Toho se dosáhne vygenerováním náhodného čísla v rozsahu od nuly do celkového počtu karet v panelu, což umožňuje, aby karty zaujímaly v daném panelu libovolnou pozici. Tento přístup zajišťuje náhodné rozložení karet a činí každou novou herní relaci jedinečnou, protože počáteční umístění karet je pokaždé jiné. Logika algoritmu je znázorněna v diagramu 5.15



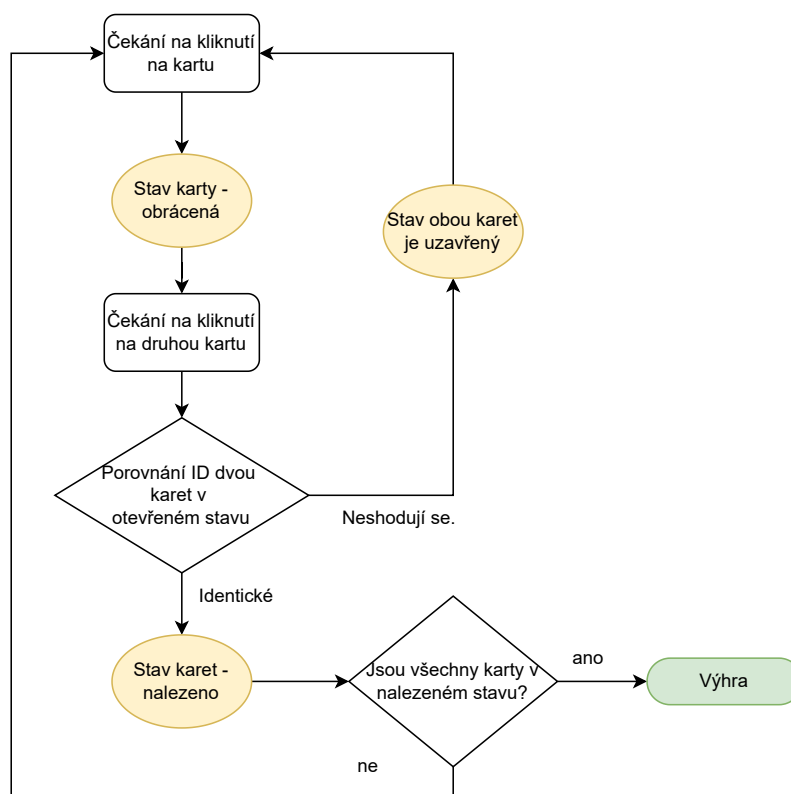
Obrázek 5.15: Na obrázku je znázorněn proces míchání karet. Diagram ukazuje klíčové kroky: načtení každé karty pomocí metody *GetChild*, vygenerování náhodného čísla pomocí funkce *Random.Range* a přiřazení nového indexu kartám pomocí metody *SetSiblingIndex*. Tento proces se opakuje pro všechny karty, dokud každá karta nemá nový index, což zajišťuje jejich náhodné rozložení.

Metoda *GetChild* slouží k přístupu k podřízeným objektům rodičovského objektu podle indexu. V Unity může mít každý objekt podřízené objekty a metoda *GetChild* umožňuje vybrat konkrétní podřízený objekt ze seznamu podřízených objektů podle jeho pořadového čísla. V algoritmu se používá k postupnému přístupu ke každé kartě, která je potomkem panelu karet. Díky tomu lze při míchání s každou kartou pracovat jednotlivě. Metoda *SetSiblingIndex* slouží ke změně pořadí objektů v hierarchii rodičovského objektu. Tato změna pořadí umožňuje řídit vizuální a logické uspořádání objektů v jejich nadřazeném kontejneru. V algoritmu se metoda používá k nastavení nového indexu pro každou kartu v hierarchii panelu karet. Náhodná hodnota indexu vygenerovaná funkcí *Random.Range* přiřadí každé kartě novou pozici, čímž se realizuje algoritmus míchání karet. *Random.Range* je funkce,

která generuje náhodné číslo v daném rozsahu. V závislosti na předaných parametrech může vracet buď celá, nebo reálná čísla.

Algoritmus pro kontrolu dvojic a logika dokončení hry

Algoritmus ověřování dvojic karet funguje následovně: při interakci uživatele s kartou dojde k jejímu „otočení“. Pokud se jedná o druhou „otočenou“ kartu, systém zkontroluje, zda se jejich ID shodují. Pokud se shodují, karty zůstanou otevřené a přehraje se tón úspěchu, který potvrdí správnou shodu. Pokud se ID neshodují, karty se vrátí do původního stavu se zpožděním, aby si hráč mohl zapamatovat jejich pozici, a přehraje se zvuk upozorňující na chybu.



Obrázek 5.16: Schéma znázorňuje proces kontroly párování karet. Začíná tím, že čeká na akci uživatele, který klikne na kartu a otočí ji. Po výběru první karty systém čeká na výběr druhé karty, poté se porovnají ID obou otevřených karet. Pokud se karty shodují, zůstanou otevřené jako „nalezené“, pokud ne, karty se vrátí do uzavřeného stavu. Proces končí kontrolou, zda byly nalezeny všechny karty, která v případě úspěchu aktivuje scénář výhry.

Logika konce hry se aktivuje, když jsou všechny karty na poli ve stavu „nalezeno“. Počítání počtu takových karet se provádí tak, že se projdou všechny položky a porovná se jejich stav s požadovaným. Jakmile jsou všechny karty nalezeny, aktivuje se vizuální indikátor dokončení úrovně a hra se po krátké prodlevě automaticky přesune na další obrazovku. Logika algoritmu je znázorněna v diagramu 5.16

5.2.4 Běžec

Hra *Běžec* je založena na dvou hlavních algoritmech: logice hráče (jeho pohyb, kolize s překážkami a bodování) a algoritmu pro generování překážek (netopýrů). Každý z nich je podrobněji popsán níže.

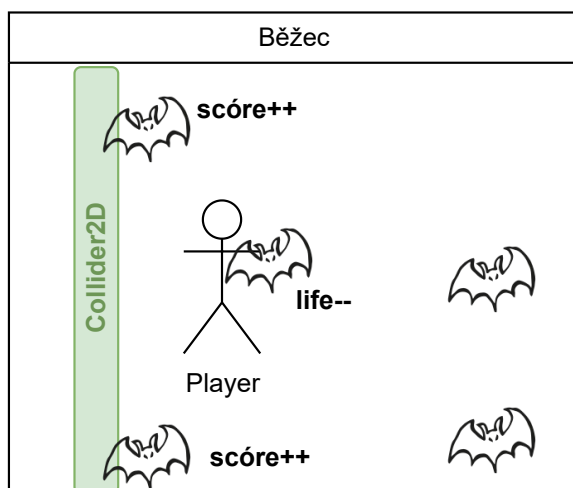
Logika hráče

Logika hráče ve hře je založena na podobných principech jako v otevřeném bludišti se sběrem klíčů. Pohyb postavy se provádí pomocí *Input.GetAxisRaw* a ošetření kolizí pomocí metody *Collider2D* a *OnTriggerEnter2D*.

Pohyb hráče nahoru a dolů po svislé ose herního světa je ovládán pomocí kláves W (nahoru) a S (dolů). Podobně jako v otevřeném labyrintu, kde se postava pohybovala po buňkách bludiště, se zde hráč pohybuje mezi konkrétními body na obrazovce. To vytváří pocit plynulého pohybu, podobně jako v předchozí hře.

Kolize hráče s překážkami (netopýry) je rovněž implementována prostřednictvím *Collider2D* a metody *OnTriggerEnter2D*. Když se hráč srazí s netopýrem, sníží se jeho zdraví a překážka (netopýr) se zničí. Pět kolizí má tedy za následek ztrátu .

Bodování ve hře provádí neviditelný *Collider2D* umístěný za hráčem a roztažený přes celou scénu. Když překážka (netopýr) projde kolem hráče a nesrazí se s ním, počítá se jako získaný bod (viz Obrázek 5.17). Počet bodů tedy odpovídá počtu proletěných překážek. Jakmile hráč získá 100 bodů, hra se považuje za ukončenou.



Obrázek 5.17: Tento diagram znázorňuje interakci mezi různými složkami herní logiky. Během hry se při setkání s určitými objekty nebo podmínkami skóre postavy zvyšuje (označeno score++) nebo snižuje počet životů (označeno life-). Prvek *Collider2D* slouží k počítání skóre ve hře (předané netopýry).

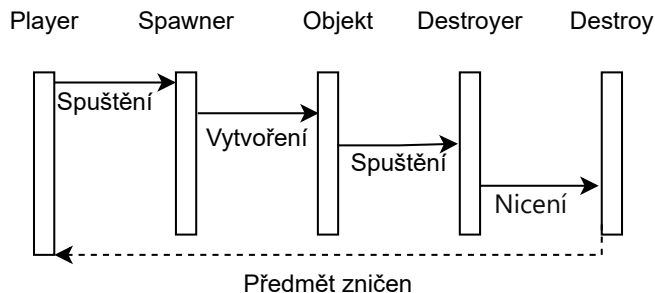
Algoritmus generování překážek

Ve hře je implementován mechanismus generování netopýrů, který je navržen tak, aby se tyto překážky objevovaly v různých částech herní úrovně. Netopýři se generují pomocí tří různých přednastavených možností (viz Obrázek 4.19).

Tento přístup využívá komponentu, která ve hře průběžně generuje netopýry. V tomto případě se z předem definované řady možných variant netopýrů (nahore a dole, nahore a

uprostřed, uprostřed a dole) vybere náhodná varianta, která se vygeneruje. Tyto varianty se od sebe liší a mohou ovlivnit obtížnost setkání. Vytváření probíhá na pevně daném místě na mapě, v pravé části obrazovky za scénou, a netopýři míří doleva směrem k hráči. Intervaly generování jsou proměnlivé: začínají v určitý čas a s postupem hráče se snižují, čímž se hra stává náročnější a napínavější.

Ve hře je také implementována logika ničení překážek, které byly překonány. Po určité době po opuštění scény jsou odstraněny, aby nedošlo k přetížení scény velkým počtem objektů (viz Obrázek 5.18).



Obrázek 5.18: Diagram ukazuje proces řízení životního cyklu objektu v herním enginu, od jeho vytvoření až po zničení. Proces začíná inicializací objektu, kdy je objekt zaveden do herní scény. Dále je objekt vytvořen na určité pozici pomocí speciálního příkazu. Po splnění svých funkcí nebo při splnění určitých podmínek je objekt podroben destrukci pomocí dalšího příkazu, který jej trvale odstraní ze hry.

V Unity hrají metody *Instantiate* a *Destroy* [7, str. 256] klíčovou roli při správě herních objektů a umožňují vývojářům dynamicky měnit herní svět v reálném čase. *Instantiate* se používá k vytváření kopií objektů, jako jsou nepřátelé, střely nebo efekty, a lze ji nastavit tak, aby umístila nový objekt na konkrétní pozici a s určeným natočením. Tato metoda se hojně používá ke generování prvků úrovně nebo náhodně umístěných objektů, což pomáhá vytvářet jedinečné a zajímavé herní situace.

V tomto algoritmu se *Instantiate* aktivně používá k vytváření netopýřů na různých místech úrovně a také k přehrávání zvukových a vizuálních efektů při interakci těchto překážek s hráčem.

Metoda *Destroy* naopak objekty ničí, čímž uvolňuje prostředky a zabraňuje únikům paměti. Tato metoda je užitečná zejména pro odstranění objektů, které se již nepodílejí na hře, jako jsou zničené struktury nebo zničení nepřátelé. *Destroy* se také používá k vyčištění paměti od dočasných objektů, jako jsou dokončené efekty a animace, což je důležité pro udržení vysokého výkonu hry.

V tomto algoritmu se *Destroy* se zase používá ke zničení netopýřů po jejich interakci s hráčem a také k automatickému odstranění objektů po uplynutí nastaveného času prostřednictvím komponenty *Destroyer*.

5.3 Úvodní menu a nastavení

V hlavní menu hry je implementována praktická logika pro pokračování poslední herní relace. V průběhu hry se každá otevřená scéna s dalšími podmínkami ukládá do *PlayerPref*.

To znamená, že když se hráč vrátí do hlavní nabídky a rozhodne se pokračovat ve hře, program si z PlayerPref přečte údaje o poslední scéně a spustí hru od této scéně. Pokud se hráč rozhodne spustit hru znovu, začne se hrát od začátku.

Kromě toho se do PlayerPref ukládají také nastavení zvuku a hudby. To znamená, že hlasitost zvuku a hudby zůstává po celou dobu hry stejná, což umožňuje hráči přizpůsobit si zvukovou stopu podle svých představ a sledovat ji v každé scéně.



Obrázek 5.19: Ukázka hlavního menu hry a okna nastavení

Kapitola 6

Uživatelské testování

Uživatelské testování hraje při vývoji hry klíčovou roli, protože poskytuje cennou zpětnou vazbu od potenciálních uživatelů o aspektech hry, které fungují dobře, a o těch, které je třeba zlepšit. Během vývoje hry bylo provedeno testování, které hodnotilo použitelnost rozhraní, zajímavost obsahu a celkový dojem ze hry. Testování probíhalo mezi různými věkovými skupinami na hlavní platformě Windows.

6.1 Metodika

Pro testování byla vybrána skupina uživatelů ve dvou věkových kategoriích: 15 až 18 let a nad 18 let. To znamená, že testování se zúčastnili středoškoláci a vysokoškoláci. Testování se zúčastnili také lidé z různých zemí. Účastníci vyplnili dotazník ve formulářích Googlu, v němž hodnotili různé aspekty hry, včetně přehlednosti rozhraní, kvality dialogů, a také vyjádřili svůj názor na touhu hrát plnou verzi hry po jejím oficiálním vydání.

Hra byla účastníkům testování zpřístupněna na Disku Google. Museli si stáhnout soubor zip, rozbalit hru a hrát podle pokynů ve hře. Účastníci nedostali žádné konkrétní úkoly, aby bylo možné zjistit, nakolik je hra intuitivní, a získat co nejupřímnější zpětnou vazbu na hru.

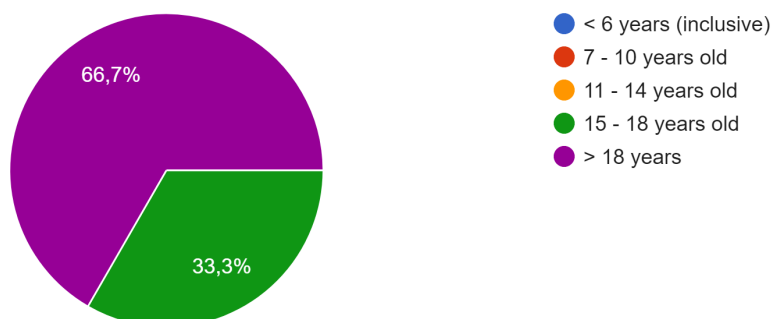
6.2 Výsledky

Dotazník vyplnilo celkem šest osob. Nízkou účast lze přičíst nucenému stažení hry z neoficiálního úložiště, nevhodnému operačnímu systému nebo obecné nechuti k formulářům. Dotazník navíc nebyl zveřejněn v komunitách, kde je vysoká pravděpodobnost „humorných“ a „ironických“ odpovědí. Při testování je třeba si uvědomit, že není přímým odrazem zkušeností uživatelů. Testeři nehrají za stejným účelem jako běžní uživatelé.

Ve formuláři pro uživatelské testování byly první otázky zaměřeny na věk a pohlaví účastníků. Tyto údaje pomáhají určit demografický profil skupiny testujících, což je důležité pro analýzu preferencí a reakcí různých kategorií hráčů. Obrázky 6.1 a 6.2 ukazují procentuální zastoupení různých typů skupin.

How old are you?

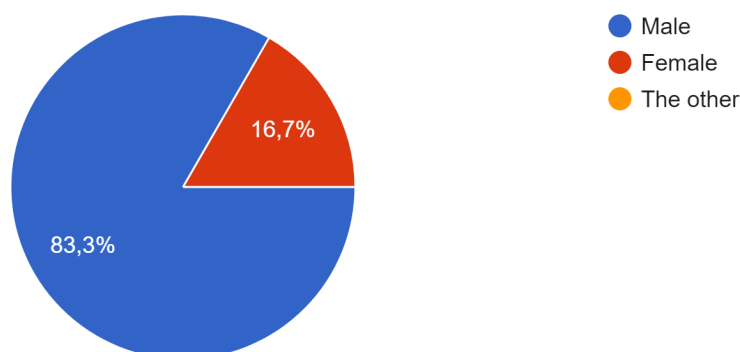
6 odpovědí



Obrázek 6.1: Graf ukazuje, že většina testujících (66,7 %) byli uživatelé starší 18 let.

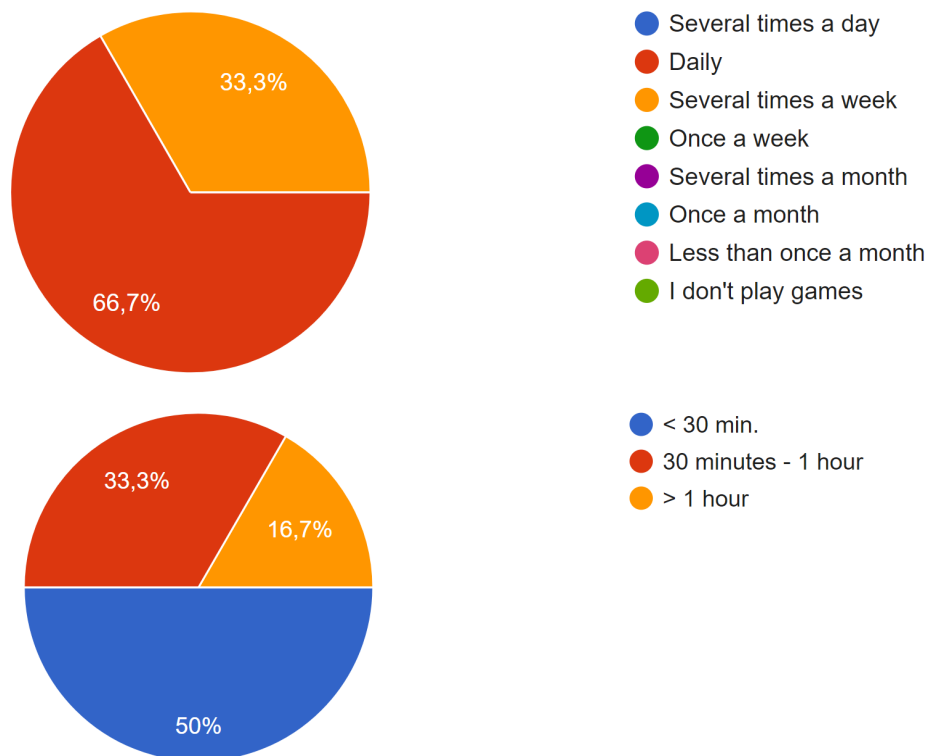
Identify your gender

6 odpovědí



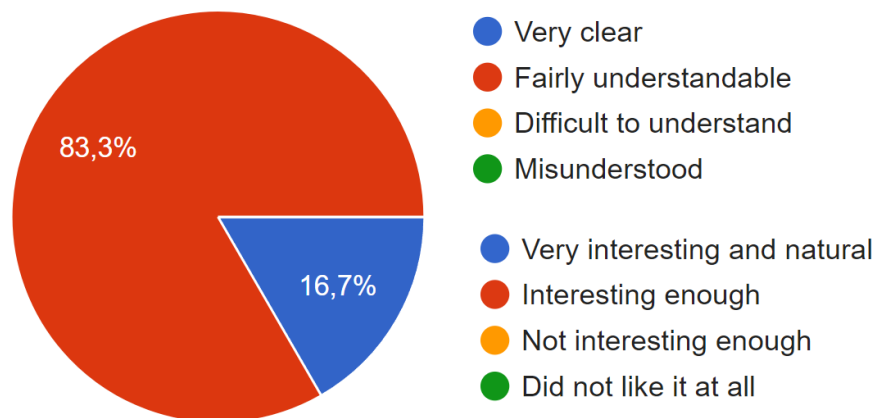
Obrázek 6.2: Z grafu vyplývá, že většinu testujících (83,3 %) tvořili muži.

Po úvodních otázkách týkajících se věku a pohlaví obsahoval formulář pro uživatelské testování otázky týkající se frekvence herní aktivity účastníků a času, který strávili hraním hry (viz Obrázek 6.3). Tyto údaje jsou určeny k analýze vlivu herních zkušeností na míru dokončení hry. Posouzení toho, jak pravidelnost herní praxe ovlivňuje dobu potřebnou k dokončení hry, poskytuje další vhled do toho, jak různé typy hráčů interagují s herním obsahem a jaké aspekty hry mohou vyžadovat dodatečné přizpůsobení pro různé skupiny uživatelů.



Obrázek 6.3: Graf v horní části ukazuje, jak často účastníci hru hrají, a graf v dolní části ukazuje, jak dlouho účastníkům trvalo dokončit celou hru.

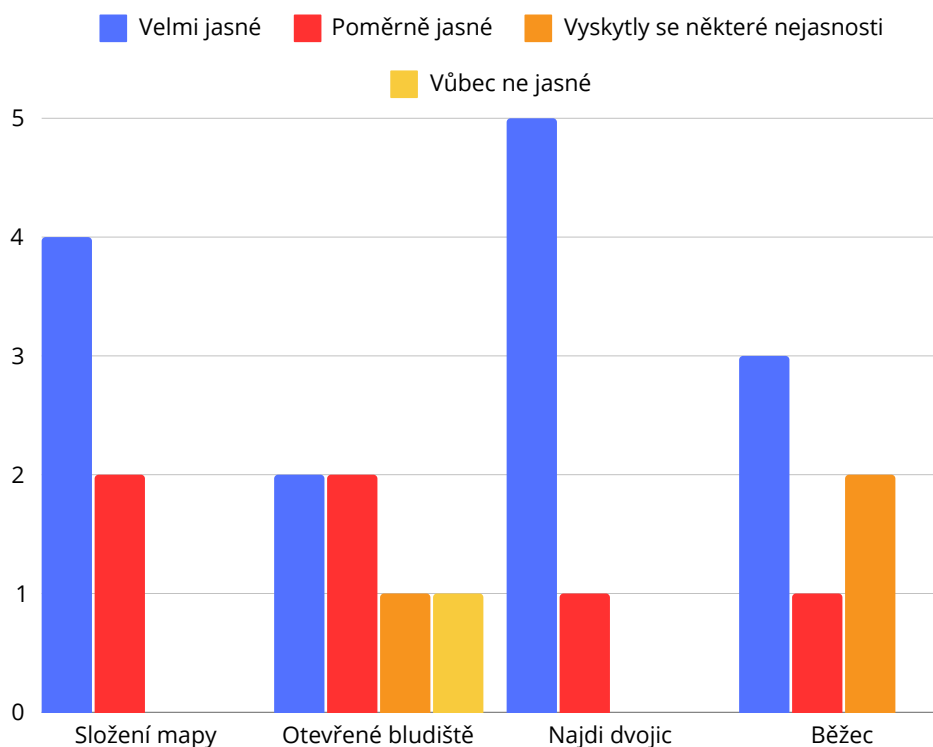
Poté se forma uživatelského testování přesunula na konkrétnější aspekty základní hry, včetně hodnocení přehlednosti rozhraní a atraktivity dialogů. Obě otázky přinesly naprosto shodné výsledky (viz Obrázek 6.4). Jeden z testujících však poznamenal, že by si přál, aby rozhraní hry bylo atraktivnější.



Obrázek 6.4: Z grafu vyplývá, že rozhraní hry je obecně přehledné a dialogy jsou poměrně zajímavé a živé.

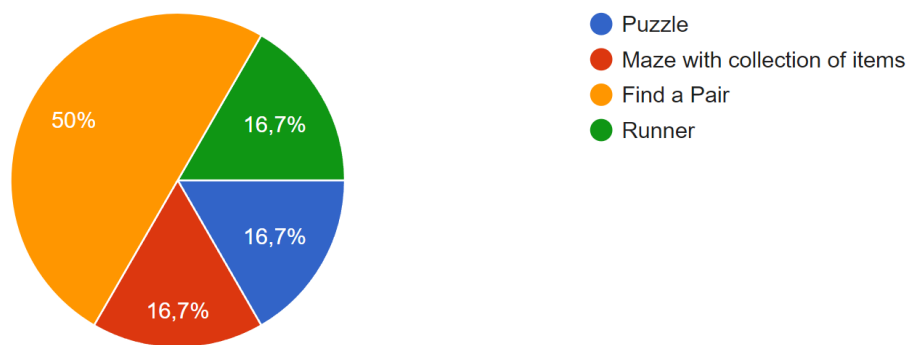
Po vyhodnocení hlavní hry přešel dotazník k rozsáhlému bloku otázek věnovaných minihrám. Každá otázka v této části byla zaměřena na jinou minihru, což testerům umožnilo podrobně zhodnotit každou z nich. Otázky se skládaly z hodnocení obtížnosti hry a hodnocení snadnosti pochopení pravidel minihry.

První sloupcový graf 6.5 ukazuje odpovědi na otázku, zda tester porozuměl ovládání a pravidlům minihry.



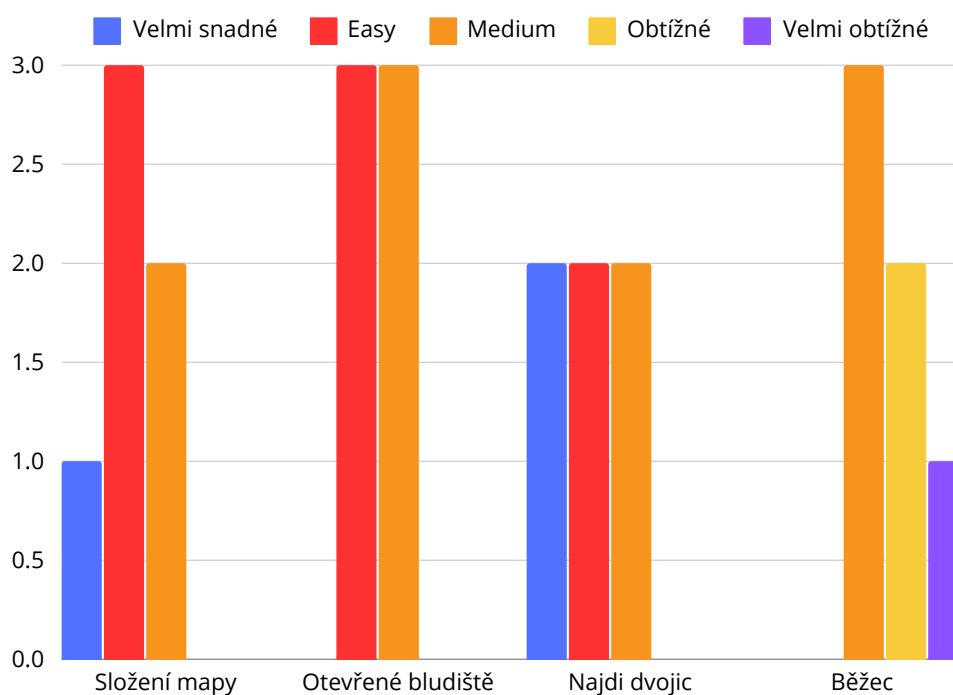
Obrázek 6.5: Tento diagram ukazuje hodnocení přehlednosti rozhraní čtyř miniher: Skládání mapy, Otevřené bludiště, Najdi pár, Běžec. Hodnocení je rozděleno do čtyř kategorií: „Velmi jasné“, „Poměrně jasné“, „Přítomny některé nejasnosti“, „Vůbec ne jasné“. Skóre jsou zobrazena na svislé stupnici, kde je každá minihra hodnocena podle těchto kategorií. Výsledky ukazují, že většina miniher je hodnocena jako přiměřeně jasná, ačkoli některé hry měly nejasnosti, které vyžadují další zdokonalení rozhraní.

Výsledky ukazují, že zavedení průvodce minihrami bylo dobrým krokem. Pokud minihra není známá nebo není dostatečně podobná původní hře, je k nim zapotřebí průvodce. U otevřeného bludiště však průvodce nestačil. Další otázka, zobrazená na obrázku 6.6, se týkala nejoblíbenější minihry.



Obrázek 6.6: Odpovědi ukazují, že velmi jednoduché minihry jsou pro testery nejvíce zajímavé.

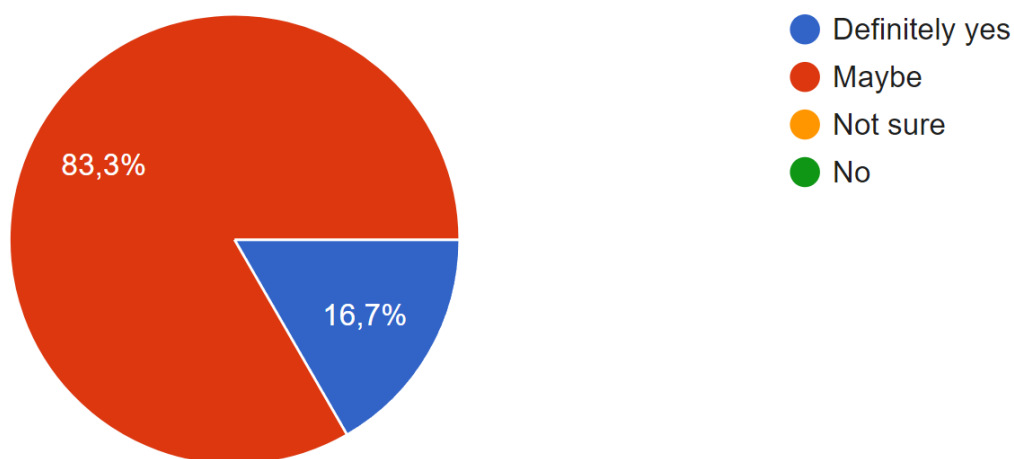
Čím je hra jednodušší, tím je více oblíbená. Minihry, co procvičují paměť, nejsou tak oblíbené. Druhý sloupcový graf 6.7 zobrazuje odpovědi na otázku ohledně obtížnosti minihry.



Obrázek 6.7: Graf v tomto souboru ukazuje obtížnost stejných čtyř minihry podle hodnocení uživatelů. Kategorie obtížnosti se pohybují od „velmi snadné“ po „velmi obtížné“. Každá minihra je hodnocena účastníky a výsledky jsou prezentovány jako sloupcový graf, kde výška sloupce odpovídá počtu odpovědí v jednotlivých kategoriích obtížnosti. Celkové hodnoty v grafu naznačují, že většina her je středně obtížná, ale některé hry jsou vnímány jako příliš obtížné nebo příliš snadné, což poskytuje příležitost pro další doladění obtížnosti.

Na základě uživatelského testování hry bylo zjištěno několik klíčových aspektů, kterým je třeba věnovat pozornost. Zaprvé byly v textech hry nalezeny gramatické chyby, což zdůrazňuje potřebu důkladnější korektury textů před konečným vydáním. Zadruhé, zpětná vazba od testerů odhalila, že manuály k minihrám **Otevřené bludiště** a **Běžec** potřebují vylepšit, aby se zvýšila jejich srozumitelnost. To zdůrazňuje význam srozumitelnosti návodu pro zajištění lepšího herního zážitku.

Kromě toho je důležitou součástí zpětné vazby i to, jaký zájem mají hráči o plnou verzi hry po jejím oficiálním vydání. Pro ilustraci tohoto aspektu je uveden graf 6.8, který ukazuje procento účastníků ochotných hrát plnou verzi.



Obrázek 6.8: Graf poukazuje na to, že testeři váhají, zda si zahrát plnou verzi hry, ale mají o ni jasný zájem.

Kapitola 7

Závěr

Cílem této práce bylo vytvořit, implementovat a otestovat detektivní počítačovou hru s prvky adventury, zasazenou do moderní mystické vesnice. Vývojový proces se zaměřil na integraci interaktivního vizuálního románu a řady miniher, které obsahovaly logické úkoly a hádanky, jež posunovaly příběh hry.

Realizace projektu ukázala, že hluboké ponoření do příběhu a aktivní interakce s herním světem mohou výrazně zvýšit zážitek ze hry. Hra úspěšně kombinuje prvky tradičních detektivních her s inovativními interaktivními přístupy a umožňuje hráčům nejen sledovat vývoj příběhu, ale také aktivně ovlivňovat jeho vyústění.

Navzdory úspěchům však byly během testování zjištěny některé nedostatky, včetně složitosti některých hádanek, které mohly odradit hráče, kteří dávají přednost dynamičtějšímu a méně inteligentnímu hernímu zážitku. To zdůrazňuje potřebu dále vyvažovat obtížnost úkolů, aby hra zůstala přístupná a přitažlivá pro širší publikum.

Na základě získaných zkušeností a zpětné vazby od prvních uživatelů se plánuje další vývoj hry. Mezi hlavní oblasti budoucích vylepšení patří optimalizace rozhraní, zlepšení grafiky a zvuku a také přidání nových příběhových linií a miniher, které ještě více vtáhnou do herního světa.

Kromě toho se bude pracovat na rozšíření přístupnosti hry, včetně přizpůsobení pro různé platformy a zařízení, což pomůže přilákat nové publikum.

Souhrnně lze říci, že hra **The Secret of Summerside: Olivia's Mysterious Journey** vykazuje značný potenciál pro zábavu i mentální cvičení a poskytuje uživatelům jedinečnou příležitost nově definovat žánr detektivních her a objevit fascinující svět dobrodružství a řešení záhad.

Literatura

- [1] BLYE, E. *The 16 Best Detective Games, Ranked? T_EXu* [online]. [cit. 2023-11-21]. Dostupné z: <https://www.thegamer.com/detective-video-games-best/>.
- [2] BOND, J. G. *From Concept to Playable Game with Unity and C#*. 2. vyd. Addition-Wesley, 2021. ISBN 978-5-4461-07-15-5.
- [3] BRAMBLE, R. *The seven stages of game development T_EXu* [online]. [cit. 2023-11-12]. Dostupné z: <https://gamedev.io/ru/blog/stages-of-game-development>.
- [4] FAHME, M.-U.-S. *How to make a game : go from idea to publication avoiding the common pitfalls along the way*. New York: Apress, 2021. ISBN 978-1-4842-6916-9.
- [5] FREECODECAMP. *What is Game Development? T_EXu* [online]. [cit. 2023-11-12]. Dostupné z: <https://www.freecodecamp.org/news/what-is-game-development/>.
- [6] GREGORY, J. *Game engine architecture*. Third edition. Boca Raton: CRC Press, 2019. ISBN 978-1-138-03545-4.
- [7] HALPERN, J. *Developing 2D games with Unity : independent game programming with C#*. New York, NY: Apress, 2019. ISBN 978-1-4842-3771-7.
- [8] KIM, S. *What is a Puzzle?* [online]. Wellesley, 2006 [cit. 2023-11-04]. Dostupné z: https://cs.wellesley.edu/~cs215/Lectures/L17-IntroGamesJigsawPuzzle/ScottKim-What_is_a_Puzzle.pdf.
- [9] KOK, B. *Beginning Unity editor scripting : create and publish your game tools*. New York: Apress, 2021. ISBN 978-1-4842-7166-7.
- [10] KUO, M.-H., YANG, Y.-L. a CHU, H.-K. Feature-Aware Pixel Art Animation. *Computer graphics forum*. Oxford: Blackwell Publishing Ltd. 2016, sv. 35, č. 7, s. 411–420. ISSN 0167-7055.
- [11] MEIER, S. a NOONAN, J. L. *Sid Meier's memoir!: A life in computer games*. W. W. Norton & Company, Inc., 2021 [cit. 2023-11-12].
- [12] MURATET, M. a GARBARINI, D. Accessibility and Serious Games: What About Entity-Component-System Software Architecture? In: *Games and Learning Alliance*. Cham: Springer International Publishing, s. 3–12. Lecture Notes in Computer Science. ISBN 3030634639.
- [13] SMITH, M. *Unity 4.x cookbook : over 100 recipes to spice up your Unity skills*. Birmingham: Packt Publishing, 2013. ISBN 978-1-84969-042-3.

- [14] TECHNOLOGIES, U. *Unity user manual 2022.3 (LTS)*. 2023. Dostupné z:
<https://docs.unity3d.com/Manual/index.html>.