

**Hodnocení:**

**Autor práce:** Václav Mašl, Matěj Černý, Filip Beneš

**Třída:** 4. I

**Vedoucí práce:** Alan Koukol

**Dne:** 26. 3. 2024

**DLOUHODOBÁ MATURITNÍ PRÁCE S OBHAJOBOU**

Téma: Úniková hra

**Vyšší odborná škola   
a Střední průmyslová škola elektrotechnická   
Plzeň, Koterovská 85**

** Vyšší odborná škola**

**a Střední průmyslová škola elektrotechnická**

**Plzeň, Koterovská 85**

Zadání dlouhodobé maturitní práce

*Zde vložte originál zadání podepsaný ředitelem školy*

Anotace

Maturitní práce se zaměřuje na interaktivní únikovou hru vytvořenou v prostředí Unity s využitím pixel artu. Hra přenáší hráče do tajemného středověkého hradu, kde se musí probojovat skrz sérii výzev, hádanek a nebezpečí, aby našel cestu ven a odhalil tajemství uvěznění.

S využitím různých herních mechanismů, jako jsou sbírání předmětů, řešení hádanek a navigace prostředím, hráči prozkoumávají jednotlivé místnosti hradu a postupují přes minihry, aby odemkli nové úseky a postoupili blíže k cíli. Vizuální stránka hry je zdůrazněna pomocí pixel artu, který zachycuje středověké prostředí a podporuje hráčovu představivost.

Celkově je projekt zábavným a poutavým dobrodružstvím, které hráčům nabízí možnost uniknout do tajemného světa středověkého hradu a prožít nezapomenutelné herní zážitky.

Klíčová slova

AI, Aseprite, Audacity, Bossfight, ColorPicker, Crafting, C#, Engine, GameObject, GitLab, Grid, Healthbar, Highlight, Hint, Hover efekt, InputField, Inventář, Item, Merge, Outline, Pausemenu, PixelArt, Prefab, Repozitář, SceneManager, Slider, Soundbar, UI, Unity

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil(a) literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

„Souhlasím s využitím mé práce učiteli VOŠ a SPŠE Plzeň k výuce.“

V Plzni dne: …..................... Podpis: ….....................................

Obsah

[Zadání dlouhodobé maturitní práce 7](#_Toc162338426)

[Anotace 8](#_Toc162338427)

[Klíčová slova 8](#_Toc162338428)

[Obsah 6](#_Toc162338429)

[Úvod 7](#_Toc162338430)

[Slovníček pojmů 7](#_Toc162338431)

[1 Vývojový deník 9](#_Toc162338432)

[1.1 Václav Mašl 9](#_Toc162338433)

[1.2 Filip Beneš 11](#_Toc162338434)

[1.3 Matěj Černý 12](#_Toc162338435)

[2 Uživatelská příručka 13](#_Toc162338436)

[Item ve scéně 13](#_Toc162338437)

[Pohyb ve scéně 14](#_Toc162338438)

[Ovládání v minihrách 14](#_Toc162338439)

[3 Části projektu 15](#_Toc162338440)

[3.1 Použité technologie 15](#_Toc162338441)

[3.1.1 Unity 15](#_Toc162338442)

[3.1.2 C# 16](#_Toc162338443)

[3.1.3 GitLab 17](#_Toc162338444)

[3.2 UI – Matěj Černý, Václav Mašl 17](#_Toc162338445)

[3.2.1 Inventář 17](#_Toc162338446)

[3.2.2 PauseMenu 19](#_Toc162338447)

[3.3 Monology – Matěj Černý 20](#_Toc162338448)

[3.3.1 Intro a Outro 20](#_Toc162338449)

[3.3.2 Hinty v první místnosti 21](#_Toc162338450)

[3.3.3 Zkontrolování meče v poslední místnosti 22](#_Toc162338451)

[3.4 Grafický design a zvuk – Filip Beneš 22](#_Toc162338452)

[3.4.1 Téma a inspirace 23](#_Toc162338453)

[3.4.2 Aseprite 23](#_Toc162338454)

[3.4.3 Místnosti 24](#_Toc162338455)

[3.4.4 Itemy 27](#_Toc162338456)

[3.4.5 Design miniher 27](#_Toc162338457)

[3.4.6 UI tlačítka 30](#_Toc162338458)

[3.4.7 Bossbattle 33](#_Toc162338459)

[3.4.8 Audacity 34](#_Toc162338460)

[3.4.9 Zvukové efekty 35](#_Toc162338461)

[3.4.10 Hudba 35](#_Toc162338462)

[3.5 Minihry – Václav Mašl, Matěj Černý 36](#_Toc162338463)

[3.5.1 Minihra – Vylomení a zlomení zámku 37](#_Toc162338464)

[3.5.2 Minihra – Runy 38](#_Toc162338465)

[3.5.3 Minihra – Souhvězdí 40](#_Toc162338466)

[3.5.4 Minihry na použití klíče 40](#_Toc162338467)

[3.5.5 Bossfight 41](#_Toc162338468)

[3.6 Implementace Zvuků – Matěj Černý 41](#_Toc162338469)

[3.7 Systém Ukládání – Matěj Černý, Václav Mašl 42](#_Toc162338470)

[3.7.1 ScriptableObject jako úložiště 44](#_Toc162338471)

[3.7.2 Změna scény 44](#_Toc162338472)

[3.8 Testování 45](#_Toc162338473)

[Závěr 46](#_Toc162338474)

[Seznam obrázků 47](#_Toc162338475)

Úvod

Vítejte v úvodu naší Únikové hry. Nejprve Vám sdělíme nějaké důležité informace o tvorbě projektu a našich myšlenkách. Podrobnější informace naleznete níže v dokumentaci.

Na začátek bychom chtěli poděkovat panu Alanu Koukolovi, za brilantní vedení a pomoc při realizaci maturitní práce.

Největší motivací pro uskutečnění projektu byla snaha vystudovat střední školu, naučit se s novými technologiemi a získat zkušenosti s týmovými projekty a prací ve skupině.

Zadání projektu jsme si vybrali předem vymyšlené, ale obohatili jsme ho vlastními nápady.

Jako hlavní přednost našeho projektu vidíme různorodost použitých technologií, miniher a robustnost vědomostí, které jsme nabili prací na projektu.

Hlavní technologií, kterou jsme využili byl herní engine Unity, ve kterém se výstupy ze všech ostatních technologií setkávaly. Další významnou technologií byl programovací jazyk C#, ve kterém byly napsány veškeré skripty použity ve hře.

Důležitou technologií pro grafický design, byl grafický editor Aseprite, ve kterém byly tvořeny veškeré vzhledové prvky hry.

Klíčovou technologií pro zvuk byl zvukový editor Audacity, ve kterém byly nahrávané a upravované veškeré zvukové stopy. Další důležitou technologií pro tvorbu hudby byl MusicFX, pomocí kterého byly tvořeny všechny hudební stopy.

Slovníček pojmů

AI – umělá inteligence

Aseprite – grafický software pro výrobu PixelArtu

Audacity – audio software pro práci se zvuky

Bossfight – souboj proti mocnému nepříteli

ColorPicker – rozhraní pro výběr barvy

Crafting – proces, při kterém kombinace itemů vytvoří jiný item

Engine – softwareový nástroj určený k vývoji videoher

GameObject – základní stavební jednotka scény, reprezentuje jakýkoliv objekt ve scéně

GitLab – software pro správu verzí

Grid – rozložení obrazu na pravidelné díly

Healthbar – ukazatel zdraví

Highlight – barevné zvýraznění objektu

Hint – nápověda

Hover efekt – efekt při přejetí kurzorem po objektu

InputField – UI objekt v Unity sloužící pro zadání textu

Inventář – kompaktní uložiště hráče

Item – předmět ve hře, který můžeme sebrat

Merge – sloučení více větví v rámci git repozitáře

Outline – barevné zvýraznění vnějšího okraje

Pausemenu – menu, které se zobrazí při zastavení hry

PixelArt – typ umění tvořený z pixelů

Prefab – unity objekt, který se dá použít vícekrát a změny se automaticky propisují

Repozitář – cloudové a lokální uložiště obsahující projekt

SceneManager – knihovna v Unity, sloužící k práci se scénami

Slider – posuvník na soundbaru určující hodnotu

Soundbar – stupnice hlasitosti zvuku

Toolbar – panel s nástroji

UI – uživatelské rozhraní

Větev (git) – alternativní verze repozitáře

# Vývojový deník

## Václav Mašl

1.Sprint

Během prvního sprintu jsem společně s ostatními pracoval na obecné domluvě práce při tvorbě projektu.

Tento proces se nám myslím povedl. Nestrávili jsme tím sice příliš mnoho času, ale vzhledem k tomu, že jsme již věděli, co od hry očekáváme, tak to bylo pouze ohraničení, co zvládneme.

Během této části nevznikla žádná část ke hře, jednalo se pouze o naše rozplánování.

Také jsme se měli čas domluvit, jak bude vypadat naše komunikace a práce s GitLabem.

2.Sprint

Tentokrát jsem měl na starost doplnění scénáře a skriptu, vytvořit popis hádanek a prostředí.

Tento úkol mě nemile překvapil. Očekával jsem, že vymyslet jednoduchou zápletku a puzzly a minihry bude snazší.

Tento úkol se na konec docela povedl, i když musím přiznat, že to byla ta nejvíce náročná část, aby se vše využilo a nebylo nic zbytečné.

Nejhorší bylo neustále vracení k předchozím místnostem, až člověku začne splývat co kam dal.

Řešením k tomuto by pravděpodobně mohlo být nějaké schéma místností a itemů, avšak na to jsem tou dobou nepřišel.

3.Sprint

Tento sprint beru jako nejméně povedený, co se týče mé práce. Za úlohu jsem měl přidat druhou místnost. To se zdálo jako poměrně jednoduché vzhledem k tomu, že již jedna místnost ve hře byla.

Velikým problémem, kterému jsem čelil bylo však, že jsem s prací v Unity byl nový. A proto když jsem měl vymyslet, jakým způsobem udělat objekt zmáčknutelný nebylo to nic snadného.

Toto mi teď, když vím, jak na to přijde jako hloupost, kterou každý vyřeší za pět minut, ale opravdu tomu tak nebylo. Nejen, že jsem měl problém s mačkáním objektů, ale také jsem poprvé pracoval s knihovnou SceneManager.

Nepříjemnost také je, že Unity je lépe připraveno na 2D hry kde se na obrazovce pohybuje postavička a většina návodů na internetu je také na hry tohoto typu.

Toto obecně znepříjemnilo mé učení Unity.

Do místnosti jsem také vložil pár objektů, a vytvořil jsem hover efekt pro sprite v této místnosti.

4.Sprint

Tento sprint byl, řekl bych, o něco povedenější než minulý.

Když se na něj kouknu retrospektivně, tak jsem vyřešil přepínání scén a přidal jsem třetí místnost. To nebyla žádná veliká výzva, vzhledem k tomu, že jsem již vše uměl z minulého sprintu.

Jediné, na co jsem zde narazil byl, problém objektů ve spritu místnosti, který se však rychle vyřešil. Jednalo se o to, že na pozadí byli předměty, které tam po dohrání minihry neměli být vůbec. Tyto menší sprity se tedy začali oddělovat od celkových spritů místnosti.

5.Sprint

Tento sprint nebylo nic zadáno učitele. Rozhodli jsme se tedy, že do hry přidám zbytek místností. Toto nebylo příliš časově náročné, avšak pár problémů se naskytlo.

Co nebylo příjemné, bylo to, že Filip, který vše musel kreslit příliš nestíhal, a proto jsem tento úkol dodělával dost na poslední chvíli. Nakonec jsme i některé sprity místností doopravovali ještě následující sprint.

6.Sprint

Znovu nic nebylo zadáno učitelem, ale měli jsme si rozmyslet co a jak abychom hru stihli během následujících třech sprintů. Proto jsme udělali dokument, kde jsme si práci rozdělili.

Mým úkolem na tento sprint bylo, abych udělal UI miniher a jejich načítání, Pausemenu a také první minihru, kterou bylo vylamování zámku. Toto bylo více práce, než co jsem dělal doteď, a proto jsem časově tolik nestíhal.

Nejdéle mi trvalo udělat systém pauzování a také pravidelný pohyb v první minihře.

Vše jsem nakonec stihl včas a vše fungovalo.

7.Sprint

Tentokrát na mém seznamu úloh bylo vytvořit dvě další minihry.

První byla minihra s runami. Dělání této minihry mě mile překvapilo. Očekával jsem totiž, že to bude náročnější, ale nakonec se ukázalo, že šlo o poměrně snadnou práci.

Další minihrou bylo spálení blokády v místnosti se schody. Tato minihra sice byla již v podstatě hotova z první místnosti, avšak v mé místnosti to nefungovalo, a proto jsem to řešil znovu.

Mezitím jsem ještě dodělával menší nedostatky na naší hře a začal jsem experimentovat s přidáním zvuků.

8.Sprint

Tento poslední sprint byl velice intenzivní nejen, že jsme museli dodělat celou hru, ale zároveň jsme museli udělat i dokumentaci.

Na hře bylo poměrně dost práce, muselo se vše dokončit a dodělat resty z minulých sprintů. Ne vše bylo příliš náročné. Z veliké většiny jsem dělal velmi podobné maličkosti v různých místnostech. Jednalo se například o předělávání itemů, změnu spritů a přepínání do a z miniher

Během těchto tří týdnů jsem na projektu strávil zdaleka nejvíce času. Také mě vývoj v tomto čase nejvíce bavil.

Sice bylo vše narychlo, ale zároveň jsem všemu lépe rozuměl a také jsem začal chápat některé vychytávky a scripty od Matěje.

## Filip Beneš

1.Sprint

V prvním sprintu jsme společně s Matějem a Vaškem promýšleli rozpoložení místností a jejich kontentu. Poté jsme potřebovali vymyslet velikost všech spritů (sprite = pixelový objekt). Finální úlohou bylo navrhnout první místnost, tedy začátek celé hry.

2.Sprint

Ve druhém sprintu jsem navrhl a vytvořil celou první místnost a k ní jsem vytvořil potřebné itemy.

3.Sprint

Ve třetím sprintu jsem navrhl a vytvořil druhou místnost, k ní jsem vytvořil potřebné itemy a vytvořil základní UI (šipky, tlačítko „inventář“ a „nastavení“).

4.Sprint

Ve čtvrtém sprintu jsem navrhl a vytvořil třetí místnost, jako u předchozích jsem k ní vytvořil i všechny potřebné itemy. Dále jsem vytvořil sprite pro první minihru.

5.Sprint

V pátém sprintu jsem navrhl a vytvořil čtvrtou a pátou místnost, k nim jsem opět vytvořil všechny potřebné itemy. Dále jsem začal s hrubými návrhy dalších miniher.

6.Sprint

V šestém sprintu jsem navrhl a vytvořil zbývající dvě místnosti, vytvořil zbylé potřebné itemy a pracoval na minihrách.

7.Sprint

V sedmém sprintu jsem udělal všechny potřebné zvukové efekty. Dále jsem dodělal zbylé potřebné UI, jako je např. úvodní obrazovka a pomocí MusicFX vytvořil hudbu.

8.Sprint

V osmém sprintu jsem dodělal všechny minihry, navrhl a vytvořil finální bossbattle a také jsem se podílel na výrobě dokumentace.

## Matěj Černý

1.Sprint

Během prvního sprintu jsem společně s ostatními pracoval na obecné domluvě práce při tvorbě projektu. Tento proces se nám, myslím, povedl.

I když jsme tomu nepřidělali příliš mnoho času, věděli jsme již, co od hry očekávat, a tak jsme si jen stanovili rámec toho, co chceme dokázat.

Během této fáze nevznikla žádná část hry; jednalo se pouze o naše plánování. Také jsme si vyhradili čas na domluvení se, jak bude probíhat naše komunikace a práce s GitLabem.

2.Sprint

Mým úkolem v tomto sprintu bylo společně s Filipem Benešem vymodelovat první místnost a následně implementovat základní GUI pro pohyb a rozhlížení hráče.

Pro mne to nebylo jednoduché, protože jsem se s enginem Unity a programovacím jazykem C# setkal poprvé.

Kromě toho jsem vytvořil několik základních skriptů pro itemy, jako je například drag and drop nebo univerzální hover efekt.

Prototyp první místnosti byl na konci tohoto sprintu hotov.

3.Sprint

V tomto sprintu jsem měl vytvořit inventář, který se bude otevírat na klávesovou zkratku. Věděl jsem, že to bude rozsáhlý úkol, a tak jsem už předem plánoval, že ho rozdělím do dvou sprintů. Zaměřil jsem se na otevírání/zavírání inventáře, reprezentaci itemů v něm (pro což jsem se rozhodl použít scriptable objecty), a také jsem řešil předávání jednotlivých předmětů mezi sloty v inventáři.

4.Sprint

Tento sprint jsem se věnoval dokončení inventáře – sbírání itemů a jejich následné vracení do scény. Dále jsem upravoval prvotní pohyb v UI vrstvě, aby byl uživatelsky přívětivější, a řešil přenášení inventáře a předmětů mezi jednotlivými scénami.

Celkově jsem se staral o merge větví v GitLabu a řešil merge konflikty.

5.Sprint

V tomto sprintu jsem se rozhodl začít rozsáhlejší část našeho projektu – ukládání mezi scénami. To mi zabralo spoustu času, protože jsem zpočátku zvolil nevhodné řešení, které jsem pak kompletně nahradil novým přístupem. I přesto, že jsem nedokázal ukládání během tohoto sprintu dokončit, začal jsem strukturovat veškeré itemy a jejich prefaby pro kompatibilitu s novým systémem ukládání. Dále jsem upravoval některé skripty, aby interagovaly s novým ukládacím systémem.

6.Sprint

V tomto sprintu jsem dokončil ukládací systém a seznámil jsem svého spolupracovníka Václava Mašla s jeho funkcemi.

Dále jsem vytvořil v inventáři crafting systém a AudioManager pro správu zvuků a hudby. Začal jsem také vytvářet výchozí třídu pro řešení dialogů, kterou jsem později využil pro třídy k řešení intra, outra, nápovědí v první místnosti a podmínek v poslední místnosti. Tento sprint byl náročný kvůli různorodosti věcí, které jsem řešil, ale nakonec se vše podařilo do konce sprintu vyřešit.

7.Sprint

Během 7. sprintu jsem dokončil intro, outro a podmínky v poslední místnosti, stejně jako nápovědy v první místnosti.

Také jsem vytvořil minihru bossfight a začal implementovat jednotlivé zvuky a soundtracky. Celkově jsem se v tomto sprintu začal věnovat kompletaci hry jako celku, abychom měli funkční hru. Dodělal jsem i synchronizaci zvuků a hudby v naší hře s panelem ovládání hudby v pauseMenu. Vzhledem k tomu, že jsme všichni postupně dávali dohromady práci, bylo potřeba řešit mnoho merge konfliktů.

8.Sprint

Poslední sprint jsem se věnoval optimalizacím hratelnosti, normalizaci zvukových efektů a hudby a celkové dokončení hry.

I přestože se vyskytly problémy s buildem naší aplikace, pracovalo se převážně bez problémů. Poslední částí tohoto Sprintu tak byla dokumentace, kterou jsme dodělali společně s Václavem a Filipem. I přesto že jsme v tomto sprintu byli v největším časovém tlaku za celou dobu vytváření našeho projektu jsme to nakonec stihli včas.

# Uživatelská příručka

I – inventář otevřít/zavřít

Esc – Pausemenu

H – hint (první místnost)

Item ve scéně

Pravé tlačítko zmáčknutí – sebrání itemu

Levé tlačítko táhnutí – pohyb itemu

Některé itemy interagují při najetí myši

Pohyb ve scéně

Najetím na kraj obrazovky – zobrazení tlačítek na pohyb v místnosti

Mezerník – přiblížení/oddálení

Ovládání v minihrách

Souhvězdí – ovládání InputField

Vylomení/Rozbití zámku – ovládáme mezerníkem

Runy – klikání levým tlačítkem myši na jednotlivé runy

# Části projektu

Obsah této části bude odpovídat konkrétnímu zadání projektu. Měl by obsahovat rozdělení celé práce na jednotlivé části včetně delegování částí na členy týmu. Musí být jasně specifikováno kdo co dělal.

## Použité technologie

### Unity

**Unity** je herní engine vyvinutý v C# a C++ společností Unity Technologies. Používá se k vývoji her pro PC, konzole, mobily a web. První verze podporovala pouze OS X.

Nám tento engine sloužil jako základ pro práci na našem projektu.

#### Hlavní vlastnosti Unity:

**Multiplatformní:** Unity umožňuje vývojářům vytvářet hry pro širokou škálu platforem, včetně Windows, macOS, Linux, iOS, Android, konzolí (PlayStation, Xbox, Nintendo Switch) a VR/AR zařízení.

**Snadné použití:** Unity je intuitivní a snadno se s ním naučí pracovat i začátečníci. Nabízí grafické rozhraní s drag-and-drop funkcionalitou, které usnadňuje tvorbu herních scén a objektů.

**Rozšiřitelnost:** Unity je vysoce rozšiřitelný engine s rozsáhlým obchodem s assety, kde vývojáři naleznou širokou škálu pluginů, modelů, textur, zvuků a dalších herních prvků.

**Komunita:** Unity má velkou a aktivní komunitu vývojářů, kteří sdílí tutoriály, tipy a rady online.

#### Výhody používání Unity:

**Zkracuje dobu vývoje:** Unity umožňuje vývojářům vytvářet hry rychleji a efektivněji díky intuitivnímu rozhraní a široké škále dostupných nástrojů.

**Snižuje náklady na vývoj:** Unity je dostupný v bezplatné verzi, která je ideální pro začátečníky a malá nezávislá herní studia. Placené verze nabízí další funkce a podporu pro profesionální vývojáře.

**Umožňuje tvorbu vysoce kvalitních her:** Unity je schopen vytvářet hry s realistickou grafikou, komplexní fyzikou a plynulou hratelností.

#### Nevýhody používání Unity:

**Náročné na učení:** I když je Unity relativně snadné se naučit, zvládnutí všech jeho funkcí a nástrojů může vývojářům zabrat značné množství času. Toto nám značně zkomplikovalo vývoj, protože před tímto projektem jsme neměli zkušenosti s herními enginy.

**Může být náročné na hardware:** Vytváření her s komplexní grafikou a fyzikou může být náročné na výpočetní výkon a paměť.

**Některé funkce jsou dostupné pouze v placených verzích:** Bezplatná verze Unity postrádá některé funkce, které jsou důležité pro profesionální vývojáře.

### C#

**C#** je moderní, objektově orientovaný programovací jazyk vyvinutý společností Microsoft. Je úzce spjat s platformou .NET Framework, která umožňuje vývojářům vytvářet aplikace pro širokou škálu platforem.

#### Hlavní vlastnosti C#:

**Jednoduchý a snadno se s ním učí:** C# je navržen tak, aby byl intuitivní a srozumitelný i pro začátečníky. Syntaxe jazyka je podobná jazyku C++, ale je zjednodušená a zbavená některých složitějších konceptů.

**Bezpečný a robustní:** C# je typově bezpečný jazyk, který pomáhá předcházet chybám a zajišťuje stabilitu a spolehlivost aplikací.

**Objektově orientovaný:** C# plně podporuje objektově orientované programování, které umožňuje vývojářům vytvářet modulární a opakovaně použitelný kód.

**Multiplatformní:** C# lze použít k vývoji aplikací pro Windows, macOS, Linux, iOS, Android, konzole (PlayStation, Xbox, Nintendo Switch) a web.

**Rozšířitelný:** C# má rozsáhlý ekosystém knihoven a nástrojů, které vývojářům usnadňují práci a rozšiřují funkce jazyka.

#### Výhody používání C#:

**Vysoká produktivita:** C# umožňuje vývojářům vytvářet aplikace rychle a efektivně díky jednoduché syntaxi a dostupnosti nástrojů pro automatizaci.

**Široká škála použití:** C# lze použít k vývoji široké škály aplikací, od jednoduchých konzolových programů až po komplexní webové a desktopové aplikace.

**Velká komunita:** C# má velkou a aktivní komunitu vývojářů, kteří sdílí tutoriály, tipy a rady online.

#### Nevýhody používání C#:

**Může být náročné na učení:** I když je C# relativně snadné se naučit, zvládnutí všech jeho funkcí a nástrojů může vývojářům zabrat značné množství času. S tímto jsme osobně neměli takový problém, neboť je jazyk z veliké části podobný Javě.

**Není tak rychlý jako některé jiné jazyky:** C# není nejrychlejší programovací jazyk a pro výpočetně náročné aplikace mohou být vhodnější jiné jazyky.

**Těsně závislý na platformě .NET:** C# je úzce spjat s platformou .NET Framework, která není dostupná na všech platformách.

#### Příklady aplikací vytvořených v C#:

* Unity
* Visual Studio
* Microsoft Office

### GitLab

**GitLab** je DevOps platforma, která pokrývá celý cyklus vývoje softwaru. Umožňuje vývojářům a týmům spolupracovat, psát kód, sledovat chyby, nasazovat aplikace a zajišťovat jejich bezpečnost – vše na jednom místě.

Nám tato platforma posloužila především tím, že jsme byli schopni pracovat zároveň na různých částech projektu. Vždy jsme pracovali v režimu, že máme vlastní git větev a po nějaké době se provedl merge těchto větví.

#### Hlavní vlastnosti GitLabu:

**Git repozitáře:** GitLab slouží jako úložiště kódu pomocí technologie Git, což umožňuje správu verzí a spolupráci na kódu mezi členy týmu.

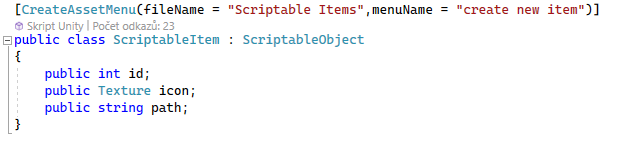
**Issue tracking:** GitLab umožňuje sledovat chyby a úkoly, což usnadňuje správu projektů a komunikaci v týmu.

**Zabezpečení:** GitLab nabízí funkce pro správu oprávnění, zabezpečení kódu a ochranu před útoky.

## UI – Matěj Černý, Václav Mašl

### Inventář

Náš inventář funguje jako pole jednotlivých slotů. Každý slot má atributy item a image kdy item je ScriptableObject který hodnotami reprezentuje daný objekt a image je RawImage, která reprezentuje Item vizuálně v inventáři.



Obrázek - ScriptableItem class

Dále je zde důležitá metoda SetValues() která zajistí promítnutí z ScriptaleItem.icon do image daného slotu. Díky tomu tak sloty můžou logicky nabývat jiných itemů než vizuálně a tyto hodnoty se pak synchronizují až touto metodou.

#### Sbírání itemů

Sbírání itemů funguje následovně. GameObject item má na sobě script Itemclass kde je přiřazený ScriptableItem odpovídající danému itemu. Při sebrání se spustí metoda PickupItem() která zničí gameObject a zároveň ScriptableObject itemu předá do inventáře. Tam je přidělen nejbližšímu volnému slotu a pomocí metody SetValues() následně zobrazen. Při sebrání je také item odebrán z úložiště scény.

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Popis byl vytvořen automaticky

Obrázek - PickupItem

#### Předávání mezi sloty

Předávání předmětů mezi sloty funguje následovně. Při započnutí tažení ze slotu se do Atributu DraggedfromSlot dosadí hodnota slotu, ze kterého je taženo. SlotClass Implementuje rozhraní IDropHandler a díky tomu můžeme za pomocí metody onDrop() získat data o upuštěném objektu. V tomto momentě tak stačí z Dropped objektu získat atribut DraggedfromSlot a máme odkaz na slot ze kterého bylo taženo i slot do kterého bylo puštěno. Pak už stačí těmto dvěma slotům jen vzájemně prohodit jejich itemy. Z tohoto důvodu jsou itemy public atributy. Pak jen opět u obou slotů zavolat metodu SetValues() a prohození už bude viditelné i vizuálně. Kromě toho se při tažení předmětu změní Parent dané Image a lze s ní pohybovat i mimo daný slot. To vytvoří hezký vizuál, kdy item následuje při tažení myš a po puštění se navrátí na svou pozici.

#### Crafting systém

Vytváření nových předmětů funguje na základě spojení dvou předmětů v jeden. To funguje na principu receptů, které jsou ScriptableObjecty.

Obsah obrázku text, Písmo, řada/pruh, snímek obrazovky

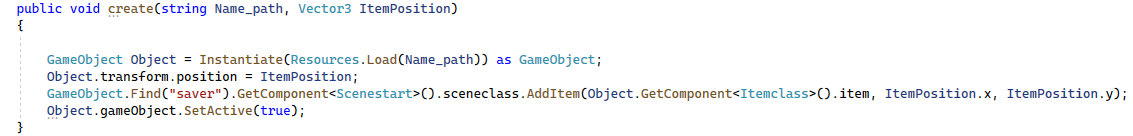
Popis byl vytvořen automaticky

Obrázek 3 - Recipe class

Crafting Systém obsahuje pole těchto receptů. Při každém položení předmětu do jednoho ze vstupních slotů se otestuje, jestli kombinace předmětů ve vstupních slotech neodpovídá nějaké kombinaci z receptů, pokud ano vytvoří ve výstupním slotu výstupní item získaný z příslušného receptu. Zároveň je ošetřeno, aby při sebrání předmětu z výstupního slotu zmizely předměty ze vstupních slotů a nevznikalo tak nekonečně mnoho předmětů. Podobným stylem jsou ošetřeny i vstupní sloty.

#### Vracení předmětů z inventáře do scény

Vracení předmětů funguje tak že v momentě, kdy uživatel pustí předmět v prostoru výstupního slotu (pravá horní část inventáře) tak je daný předmět inicializován ve scéně z prefabu za pomocí metody Instantiate() kdy cestu na daný Item získá ze ScriptableItemu který získá přes onDrop() obdobě jako při předávání předmětů mezi sloty. Pak se jen tomuto předmětu nastaví pozice, na které byl upuštěn. A je přidán do uložiště scény.



Obrázek - metoda create

### PauseMenu

PauseMenu jsme dělali tak, že jsme do objektu, který se nacházel v DontDestroy vložily nový Canvas, na který jsme vložili tlačítka a ovládání zvuku. Tento objekt se spouštěl pomocí jednoduchého skriptu, který pouze spustil zobrazení daného Canvasu a kameru posunul, aby nebylo možno manipulovat s ostatními objekty na scéně.

Součást Canvasu jsou dvě tlačítka, kde pro každé je vytvořena vlastní metoda, která se spustí při zmáčknutí.

První metodou je jednoduché opuštění hry.

Druhá metoda se stará o navrácení zpět do hry a také zajištuje že se hra dá zastavit i tlačítkem “Esc”.

Propojení slideru na hlasitost zvuku je vysvětlena v [4.7 Zvuky.](#_￼Zvuky￼)

Na tlačítka nám následně stačilo jen dodat potřebné sprity.

Obsah obrázku text, snímek obrazovky, Počítačová hra, PC hra

Popis byl vytvořen automaticky

Obrázek - ukázka PauseMenu

## Monology – Matěj Černý

### Intro a Outro

Intro i Outro jsou potomky třídy hintSystem u které jsou oba nuceni přepsat abstraktní metodu dialogEnd().HintSystem nám zajištuje načtení textového souboru a jeho předělání na pole textových řetězců se kterými pracujeme, dále nám metoda setStage() zajištuje nastavování začátků, konců a délky jednotlivých oznámení. Dále také obsahují IEnumeratory animace výpisu a přepínání jednotlivých informací.

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Popis byl vytvořen automaticky

Obrázek - metoda animatedText

Intro ještě umožnuje přeskakování a vracení se mezi jednotlivými částmi výkladu příběhu. To zajištují metody skip() a rollback(). Outro zase po ukončení monologu dá hráči na výběr, jestli chce hru opustit anebo si ji zopakovat.

### Hinty v první místnosti

HintGiver je také potomek třídy hintSystem. Jeho komunikace se však závislá na jednotlivých stavech první místnosti a nápovědy se zobrazí pouze po jejich vyvolání klávesou H. Zároveň je také ošetřeno aby v momentě vypisování nemohl uživatel vyvolat opětovné vypisování klávesou H. Ze stejného důvodu se také hned při nabití nového stavu který posune progres nápověd také není hned nastavený nový záchytný bod ale je přidán do fronty metodou setNextHint(), záchytný bod je tak předán až po ukončení monologu nápověd a pouze pokud posune progres nápověd směrem dopředu.

Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky

Obrázek - metoda setNextHint

Obsah obrázku text, snímek obrazovky, řada/pruh, Písmo

Popis byl vytvořen automaticky

Obrázek - metoda dialogEnded

### Zkontrolování meče v poslední místnosti

V poslední místnosti je podmínkou pro utkání s nepřítelem mít nabroušený meč. Pokud hráč meč vůbec nemá nebo ho nemá nabroušený bude na to patřičně upozorněn. Proto je třída swordcheck dalším potomkem třídy hintgiver aby mohla hezky a přehledně vypisovat patřičné hlášky a zároveň měla přehled o ukončení monologu. Nejdůležitější je pro tuto třídu ovšem je metoda checkInventory, která proskenuje Inventář hráče a pokud najde meč nebo naostřený meč příslušně zareaguje a zároveň ukončí vyhledávání. Tato metoda se zároveň volá jen při načtení místnosti nebo v momentě kdy by uživatel mohl předělat meč na naostřený meč, aby se zbytečně nevolala při každém pokusu o průchod dveřmi.

Obsah obrázku text, snímek obrazovky, Písmo, displej

Popis byl vytvořen automaticky

Obrázek - metoda checkInventory

## Grafický design a zvuk – Filip Beneš

Grafický design a zvuk jsou klíčovými prvky téměř každé hry či aplikace, proto se jejich tvorba nesmí zanedbat. V této kapitole se podíváme na jednotlivé fáze tvorby grafických prvků a zvukových efektů. Všechen pixel art byl tvořen v aplikaci Aseprite, o kterém se něco dozvíte v kapitole „Aseprite“. Zvuky byly nahrávány do zvukového editoru Audacity, kde se následně i poladily. Hudba byla vytvořena pomocí umělé inteligence MusicFX, ke které se v kapitole „Hudba“ budu také zmiňovat.

### Téma a inspirace

Když jsme si vybrali téma této maturitní práce, byli jsme téměř hned dohodnutí, že celá tématika hry bude inspirována středověkem. Jelikož her se středověkým tématem existuje mnoho, nebyl problém sehnat referenční obrázky, které nám výrazně usnadnily tvorbu jednotlivých místností a jejich rozmístění. Největší inspiraci jsme čerpali z české videohry „Kingdom Come: Deliverance“ nebo obrázků na internetu, které jsem v krajní nouzi použil jako referenci.

Obsah obrázku budova, schody, zeď, dveře

Popis byl vytvořen automaticky

Obrázek - inspirace

### Aseprite

Pixel art je forma digitálního výtvarného umění, která se zaměřuje na vytváření obrázků pomocí pixelů. Pro jeho tvorbu na trhu existuje mnoho grafických editorů (např.: GraphicsGale, Pyxel Edit atd.). Já osobně mám zkušenosti s Adobe Photoshop, kde se pixel art tvoří také poměrně dobře.

Pro tento projekt jsem si však řekl, že zkusím Aseprite. Na rozdíl od Photoshopu je určený výhradně pro pixel art, a tak je v něm mnoho uživatelsky přívětivých vychytávek, které vám usnadní a zpříjemní zdlouhavé repetitivní práce, jako je např. tvoření vícebarevné kostkované sítě.

Práce v Asepritu není příliš náročná. Po otevření nového projektu a vybrání rozlišení, se vám otevře přehledná kreslicí plocha, kde se vše tvoří.

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, software

Popis byl vytvořen automaticky

Obrázek - ukázka prostředí Aseprite

Jak můžete na obrázku č. 11 vidět, na levé straně máte paletu barev, kde si můžete dle svých potřeb měnit, průhlednost barvy a použít color picker pro přesnější odstín. Dole se nachází vrstvy a možnosti přehrávání, které můžete uplatnit při tvoření animací. Na pravo se nachází toolbar, ten je však potřeba pouze v případě, že chcete změnit mód jednotlivého nástroje nebo si nepamatujete zkratky. Nahoře je lišta s otevřenými projekty. Ještě nad lištou můžete najít možnosti pro úpravu celého projektu, tam si můžete např. upravit rozlišení, otočit plátno a mnoho dalších pomůcek.

Aseprite si můžete zakoupit na platformě Steam přibližně za 17€.

### Místnosti

Místnosti jsou klíčovým prvkem naší hry, jelikož je hráč bude prozkoumávat jednu po druhé a hledat způsob k postoupení do další místnosti. Z tohoto důvodu jsou nejdůležitějším pixel artem, nad kterým jsem se vždy musel dlouho zamyslet. Teprve pak mě čekal zdlouhavý proces stálého poupravování a dolaďování, který se mohl opakovat tak dlouho, dokud jsem nebyl spokojený s finálním vzhledem.

Každá místnost má rozměry 569px na 160px, hráč ale nikdy nevidí celou místnost. Vidí zhruba 3/5 místnosti. Proto jsou místnosti navrženy tak, aby se hráč musel v místnosti rozhlížet a hledat interaktivní věci. Postup tvorby je následující.

První věc, kterou jsem vždy začal, bylo hledání referenčních obrázků a rozplánování celé plochy.

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, text

Popis byl vytvořen automaticky

Obrázek - návrh místnosti

Na obrázku č. 12 můžete vidět způsob, kterým jsem prvotní rozplánování dělal.

1. Začal jsem tvořením jednoduchého gridu. (šedá barva)
2. Pokračoval jsem vyznačením rohů místnosti. (žlutá barva)
3. Dále jsem si vytvořil jednoduchý outline místnosti. (oranžová barva)
4. Po-té bylo potřeba vyznačit vchody a východy. (červená a růžová barva)
5. Na konec jsem si pro pohodlí vyznačil zakřivení podlahy. (růžová barva)

Berte na vědomí, že každý z těchto bodů je vlastní vrstvou, proto nevadí překrývající se barvy. Viditelnou budete mít pouze jednu z těchto vrstev a to tu, se kterou zrovna pracujete.

Druhou věcí, kterou chcete pokračovat jsou podlaha a zdi. Osobně doporučuji začínat podlahou, protože se dělá nejsnadněji, a při pozdější tvorbě zdi tak můžete rovnou vidět, jak ji rovnou přizpůsobit vůči podlaze. V tomto kroku bych doporučil nedělal vše úplně do detailů, jelikož po dokreslení celé místnosti se vám zaplní hodně místa, a vy tak nemusíte detailovat věci, které stejně nebudou ve finále ani vidět.

Třetí krok, který doporučuji, je udělat všechny dveře, vchody, schody, zkrátka všechno, co zabírá více místa. Jako u předchozího bodu, tyto objekty nemusí být extra vydetailované, jelikož to je nejlepší udělat až na konci.

Čtvrtá věc, která vás čeká je dodělání všech menších objektů (pochodně, židle atd.). Ty už můžete rovnou plně nastylovat, protože v dalším kroku vše doděláte.

Pátým krokem je dodělání všech detailů. Tím myslím stíny, odlesky, praskliny atd. Vše by mělo začít vypadat dobře a vy se cítit dobře. Všechna předchozí dřina se vyplatila.

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, PC hra

Popis byl vytvořen automaticky

Obrázek - místnost 2

Šestým a posledním krokem je tvorba tzv. highlightů. Highlight je zvýraznění, které chcete udělat na každé věci, se kterou bude uživatel schopný interagovat. Toto zobrazení se zobrazí pouze, když bude mít uživatel kurzor na daném objektu. Já jsem highlight pojal tím způsobem, že jsem celý objekt upravil tak, aby působil, jako že se rozsvítil. Každý to však může pojmout jinak, další nápady, které jsem měl, byly např. vyskočení otazníku na objektu a přehrání jednoduché třesoucí animace.

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, PC hra

Popis byl vytvořen automaticky

Obrázek - ukázka highlight

Na obrázku č. 14 můžete vidět ukázku highlightu.

### Itemy

Itemy jsou jedny z vedlejších prvků naší hry, které slouží výhradně jako součást, nebo klíč k nějaké minihře. Itemy jsou rozházeny mezi všemi místnostmi, a některé se dají získat pouze pomocí kombinace dvou jiných v inventářovém oknu.

Designování itemů byl spíše takový odpočinek, naproti dělání velkých místností, protože rozlišení všech itemů je 32px na 32px. Avšak i přes to jejich tvorba není úplně beze snahy. Menší plocha sice znamená méně práce, ale více plánování (v poměru velikosti), jelikož na menší ploše se tvary vyjadřují pomocí pixelů znatelně hůře.

Jako vždy bych začal vyhledáním nějakého referenčního obrázku. Dále si vyznačte délku a šířku itemu, který budete kreslit. Vytvořte přesnou siluetu. Vybarvěte, nastínujte a máte hotovo. Pokud to je nějaký složitější item, doporučuji si dělat vrstvy.

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, text

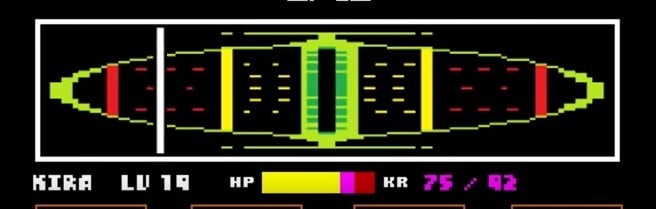
Popis byl vytvořen automaticky

Obrázek - item torch

### Design miniher

Minihry jsou v únikové hře velice důležitým elementem, dodávají celé hře trochu variace od stálého prozkoumávání a hledání. Není pouze důležité, aby dobře fungovaly, ale také aby dobře vypadaly a zaujmuly hráče.

Minihry, na rozdíl od místností nebo itemů, jsou originálnějším výtvorem. U všech se snažím, aby byly jedinečné a aby nevypadali, jako něco, co jste mohli vidět v nějaké jiné hře. Proto jsem při jejich tvorbě nepoužíval žádné reference, někdy jsem se inspiroval funkcí nějakých miniher z her, které jsem hrál.



Obrázek - inspirace minihry

Jak můžete vidět na obrázku č. 16. Pro minihru, kterou vám zde ukážu, jsem se inspiroval hitbarem ze hry Undertale. Jde v něm o zasažení středu pro maximální efekt.

V naší minihře se snažíme pomocí podobného způsobu rozbít zámek, vytvořil jsem ho následovně.

Při tvorbě minihry na rozlišení příliš nezáleží, protože v unity se to ještě dá doupravit podle potřeby, tak jsem zde zvolil 200px na 100px. Zámek budeme rozbíjet v pěkné místnosti, ne v žádném žaláři, tak jsem zvolil luxusně působící design.

Prvně jsem vytvořil siluetu inspirovanou luxusním nábytkem, kterou jsem vybarvil jednou barvou a nahoru do středu jsem nakreslil emerald, jenž přidává efekt drahoty.

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, Počítačová hra

Popis byl vytvořen automaticky

Obrázek - návrh minihry

Druhým krokem bylo vytvoření stupnice, která se nachází ve středu. K tomu jsem rovnou vše nastínoval a provedl potřebné finální úpravy. Vzhled okrajů je inspirován vyleštěným dřevem a je vytvarován podobně jako dveře, které se v této místnosti nachází. Stupnice ve středu má povrch inspirován zpracovanou kůží a na ní jsem vyznačil nejefektivnější místo červenou barvou. Její střed se nachází pod emeraldem.

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, text

Popis byl vytvořen automaticky

Obrázek - minihra rozbití zámku

Posledně zbýval vytvořit lokátor, který musíme zastavit na středu. Snažil jsem se, aby vypadal jako drahý šperk. Okraje má rubínové a střed safírový. Tento lokátor jsem potom musel zvlášť exportovat, aby se sním dalo v Unity hýbat

Obsah obrázku snímek obrazovky, text, Multimediální software, Grafický software

Popis byl vytvořen automaticky

Obrázek - tečka

Kouzlo miniher spočívá v tom, že na vás mohou vyskočit v jakékoliv velikosti. Ukážu příklad ještě jedné, kde inspirace plynula ze severské tématiky. Proto jsem zvolil robustní, kamenný vzhled, do kterého musíme vytesat správné runy. Význam run, které jsem zvolil, nemá žádnou hlubokou logiku. Zvolil jsem pouze runy, které měli nějaký šlechetný význam, jako např. znak „fehu“, který znamená bohatství (následující obrázek, prostřední znak).

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, Počítačová hra

Popis byl vytvořen automaticky

Obrázek - minihra runy

### UI tlačítka

User interface je jedna z nejdůležitějších aspektů jakékoliv aplikace. Uživatelé se s jeho pomocí pohybují po virtuálním prostředí, které jsme pro ně vytvořili. Dává tedy smysl, že jeho vzhled je také důležitý. Z tohoto důvodu se vzhled vašeho UI nemůže zanedbat. Ze všech tlačítek by mělo vyplívat, co dělají a k čemu slouží.

UI by se dalo rozdělit do mnoha typů, a tak vás s nějakými seznámím, ukážu tvorbu a výsledný produkt.

Začal bych první věcí, kterou uživatel vidí. Titulní obrazovka je první věc, kterou hráč uvidí po spuštění hry a má za úkol přilákat pozornost, vyvolat zájem a představit hráči základní informace o hře (téma, místo, čas).

Při její tvorbě jsem hledal, jako vždy, referenční obrázky, jako jsou hrady, lesy, hory, titulní obrazovky jiných her atd... Vzhled, který jsem zvolil, je osamělí hrad v lese s výhledem na hory a jezero. V popředí je smrkový les a v pozadí je polojasná noční obloha, na které jsou hvězdy a plný měsíc.

Obsah obrázku Multimediální software, snímek obrazovky, Grafický software, text

Popis byl vytvořen automaticky

Obrázek - title screen

Zvolil jsem tento vzhled pro vyvolání pocitu vzrušení a proto, že se na to příjemně kouká. Je z něj také poznat, že se hra odehrává ve středověkém hradu a v noci.

Další UI položkou je herní menu. Herní menu je centrálním bodem, kde hráči interagují s herními možnostmi a nastaveními. Poskytuje jim přístup k různým funkcím hry a umožňuje jim ovládat průběh a zážitek ze hry.

Tyto tlačítka mohou být velká, a tak jejich tvorba není příliš těžká. Stačí vytvořit slibný tvar, ten vybarvit a vystínovat a napsat na něj jeho funkci.

Design našich tlačítek je inspirován dobou, ve které se hra odehrává (středověk). Kvůli tomu jsem designoval dva typy tlačítek, jedny s dřevěnou tématikou a druhé s kamennou tématikou.

Obsah obrázku text, snímek obrazovky, Multimediální software, Grafický software

Popis byl vytvořen automaticky

Obrázek - návrh tlačítka exit

Dále se zmíním o tlačítkách ovládání. Tlačítka ovládání je široká škála tlačítek, sloužících pro nejrůznější účely. Tato tlačítka by měla být co nejjednodušší, malá, jednoznačná, srozumitelná.

Prvním typem ovládacího tlačítka jsou ovládací šipky. Tyto šipky slouží k otáčení v místnosti. Jejich design má připomínat ohnutý plech.

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, software

Popis byl vytvořen automaticky

Obrázek - ovládací šipka

Dalším ovládacím tlačítkem, které bych chtěl představit, je soundbar. Soundbar jsem designoval s myšlenkou v hlavě, že má být v levém dolním rohu, má jednoznačně ukazovat, jak sním manipulovat a to, že ovládá zvuk. Nakreslil jsem tedy hlavu megafonu, ze které vede čára, na níž sedí pointer, který se dá posouvat. Celkový vzhled je opět inspirován dřevem a kamenem.

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, software

Popis byl vytvořen automaticky

Obrázek - soundbar

Dalším druhem UI je inventář. Inventář je velice důležitým elementem her. V inventáři se dají zjistit nějaké informace o jednotlivých itemech, skombinovat je, nebo je uschovat a následně se podívat, které itemy vlastníme.

Samotný inventář je navržen tak, aby z něj jednoznačně vyplývalo, jak ho použít. Na levé straně je grid okének, kde se budou skladovat itemy. Na pravé straně je okno detailních informací a tři okénka. Dvě horní okénka slouží pro kombinaci dvou itemů, spodní slouží pro výsledek jejich kombinace. Samotný design je udělaný tak, aby vypadal, jako kůže rozvinutá na zemi. Na ní jsou kamenné destičky, které znázorňují prázdné item sloty. V pravém horním rohu se nachází papír, který bude zobrazovat informace.

Obsah obrázku snímek obrazovky, Multimediální software, Grafický software, software

Popis byl vytvořen automaticky

Obrázek - inventář

### Bossbattle

Bossbattle je součást hry, kde se střetnete s mocným nepřítelem. V našem případě se střetneme s tím, kdo nás uvěznil a utečeme. Je to tedy takové zavrcholení celé hry. Udělat bossbattle v únikové hře, kde je hlavním úkolem procházení místnostmi je celkem těžké. Proto jsme se inspirovali starými Pokemon hrami, které vycházely na herní konzole Game Boy.

Princip boje je „kámen, nůžky, papír“, proto jsem navrhl tři tlačítka, pro tři typy rozhodnutí (seknutí mečem, obrana štítem a úder pěstí). Zároveň budeme schopni přežít čtyři rány, to platí i pro nepřítele, proto jsem vytvořil jednoduchý healthbar, pro indikaci aktuálního stavu.

Dalším důležitým aspektem je prostředí. Jako v Pokemon hrách jsem vytvořil dvě travněná pole, na kterých stojíme. Na kole naproti nám se nachází nepřítel. Pozadí celého bossbattlu je jednoduchá modrá tapeta s gradientem.

Pro design nepřítele jsem zvolil vzhled středověkého šlechtice. Dal jsem štít a kopí, aby dávalo smysl, že může útočit a bránit se stejně jako my. Má šedé šaty, hnědý plášť, tmavě hnědé kalhoty a černé boty.

Pro náš design jsem zvolil vzhled vězněného šlechtice. Máme, stejně jako nepřítel, hnědé vlasy a oči. Špinavé šedé triko s dlouhými rukávy. Jsme vychrtlí a držíme meč, který jsme sebrali v šesté místnosti.

Obsah obrázku snímek obrazovky, Multimediální software, PC hra, Grafický software

Popis byl vytvořen automaticky

Obrázek - bossfight

### Audacity

Audacity je zvukový editor, který si můžete zdarma stáhnout z internetového prohlížeče a provádět na něm jednoduché zvukové úpravy.

Audacity umožňuje:

1. Nahrávání zvuku: Audacity umožňuje nahrávat zvuk z různých zdrojů, jako jsou mikrofony, přehrávače, nebo dokonce zvukové stopu z videa.
2. Úpravy zvuku: Program nabízí různé možnosti úprav zvuku, včetně střihu, kopírování, vkládání, změny hlasitosti, filtrování, změny rychlosti a pitche, odstranění šumu a mnoho dalšího.
3. Efekty: Audacity obsahuje mnoho efektů, jako jsou reverb, ekvalizér, zkreslení, kompresor, a mnoho dalších, které můžete použít pro vylepšení nebo změnu zvuku.
4. Podpora formátů: Podporuje širokou škálu zvukových formátů, včetně WAV, MP3, AIFF, FLAC a mnohé další.
5. Multi-platformní podpora: Audacity je k dispozici pro různé operační systémy, včetně Windows, macOS a Linux.

Na následujícím obrázku můžete vidět editovací prostředí a soundtrack titulní obrazovky.

Obsah obrázku text, snímek obrazovky, displej, software

Popis byl vytvořen automaticky

Obrázek - Audacity

### Zvukové efekty

Zvukové efekty jsou klíčovým prvkem vytvářejícím atmosféru a prohlubujícím herní zážitek v jakékoli hře. Zvukové efekty mohou hráče vtáhnout do herního světa, poskytnout jim zvukovou zpětnou vazbu o jejich akcích a událostech ve hře a přidat do hry další rozměr emocí a realismu.

Pro realizaci zvukových efektů jsem přes mikrofon mých sluchátek nahrával zvukové stopy do audio editoru Audacity.

V Audacity jsem zvukové stopy zesílil, potlačil jsem šum, sestřihal a exportoval do souboru mp3.

### Hudba

Hudba je dalším důležitým prvkem v každé hře, který může vytvářet emocionální atmosféru, posilovat herní zážitek a podporovat dramatické momenty ve hře. Dobře vybraná hudba může hráče vtáhnout do herního světa a prohloubit jejich zapojení do příběhu a prostředí.

Bohužel nejsem příliš nadaný skladatel, a tak jsem pro tvorbu hudby použil AI Test Kitchen.

AI Test Kitchen je umělá inteligence od společnosti Google, kde lidé mohou vytvářet nejrůznější věci pouze pomocí textového popisu. Já jsem používal konkrétně MusicFX.

MusicFX je schopen z jednoduchého promptu vytvořit hudební skladbu, která je na celkem obstojné úrovni. Vytvořil jsem v něm celkem 3 skladby. První jsem vytvořil hudbu pro titulní obrazovku. Po-té jsem vytvořil hudbu pro bossbattle a nakonec hudbu pro procházení celé hry. Prompt pro první skladbu jsem napsal následovně:

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky

Obrázek - MusicFX

Soundtrack, který jsem dostal bohužel nebyl perfektní, a tak jsem provedl minimální změny v Audacity. MusicFX je velice mladý, a tak se od něj ještě nedá očekávat perfektní hudba, věřím ale, že v budoucnu dosáhne vysoké úrovně.

## Minihry – Václav Mašl, Matěj Černý

Vytváření miniher se lišilo minihru od minihry. U miniher, které se otevírají jako nová scéna je podobný základní kód. Který řeší, co se má stát při zavření minihry nebo při jejím splnění.

Obsah obrázku text, Písmo, řada/pruh, snímek obrazovky

Popis byl vytvořen automaticky

Obrázek - spouštění miniher

Jedná se o část, ve které znemožníme uživateli používat inventář a vycentrujeme kameru, aby byl záběr vycentrován.

Veškeré minihry také obsahují pozadí. Pozadí je v každé minihře stejné

### Minihra – Vylomení a zlomení zámku

Obsah obrázku snímek obrazovky

Popis byl vytvořen automaticky

Obrázek - ukázka Zlomení zámku

Tuto minihru jsme tvořili pomocí skriptu, který zajistí neustálý pohyb ukazatele po baru. Poté co se hráč zmáčkne mezerní hra zkontroluje, zda se trefil do určené části baru, pokud ano přičítá se mu skóre, hráč se musí trefit čtyřikrát.

Nápověda k ovládání a výpis skóre jsme řešili pomocí UI.

Ovládání potřebného skóre, či kam se hráč musí trefit se dá jednoduše změnit přímo ve skriptu.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obrázek - ukázka dotMovement

Druhá varianta této minihry funguje naprosto stejně pouze se zde nachází jiná podoba baru.

### Minihra – Runy

Obsah obrázku snímek obrazovky, 3d modelování

Popis byl vytvořen automaticky

Obrázek - ukázka minihry Runy

Při vývoji této minihry jsme použili práci se soubory. A v minihře samotné jde především o to, aby se nastavila správná kombinace run, která je vidět v jiné místnosti.

Na začátku kromě běžného skriptu, nastavíme obrázky na první z nich tímto kódem.

Obsah obrázku text, Písmo, snímek obrazovky, bílé

Popis byl vytvořen automaticky

Obrázek - nastavení spritů

Následně máme pro každé tlačítko vlastní metodu a vždy se zkontroluje, zda je správná kombinace a pokud ne změní se obrázek.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obrázek - ukázka změni spritu a ověření

U této minihry se veškeré obrázky nahrávají přímo v unity.

Obsah obrázku text, snímek obrazovky, software, číslo

Popis byl vytvořen automaticky

Obrázek - ukázka vložení spritů v Unity GUI

### Minihra – Souhvězdí

Obsah obrázku klipart, Animace, snímek obrazovky, Kreslený film

Popis byl vytvořen automaticky

Obrázek - ukázka Souhvězdí

Tuto hru jsme vyrobili velice jednoduše, a to že na předpracovaný sprite se položí InputField, ze kterého následně čteme v kódu a ověřujeme správnost hesla.

Obsah obrázku text, řada/pruh, Písmo, snímek obrazovky

Popis byl vytvořen automaticky

Obrázek - ukázka kontroly hesla

Tento kód se spustí, jakmile hráč odklikne InputField.

### Minihry na použití klíče

Těchto miniher je ve hře hned několik. Nachází se již v první místnosti, a na stejném principu funguje také spálení blokády na schodech.

Udělali jsme skript, který se vloží na objekt, který bude sloužit jako klíč. Tento skript se spustí pokaždé když hráč skončí kliknutí na tlačítku myši. A ten ověří, jestli se daný předmět nachází na objektu, který slouží jako klíčová díra. Pokud tomu tak je spustí se metoda, která do hry přidá tlačítko na průchod a odebere danou klíčovou dírku a předmět, který sloužil jako klíč.

### Bossfight

Tato minihra vychází z klasické hry kámen nůžky papír. Zvolením jedním z zbraní se zvolí jedna z výčtu hodnot choices a následně se zavolá metoda pro vyhodnocení výsledku. Nepřítelovo zbraň se zvolí náhodně nezávisle na hráči. Po vyhodnocení se příslušné entitě (hráči / nepříteli) odečte život, pokud nedošlo k remíze a celý průběh tohoto střetu se vypíše do příslušného pole . Po celou dobu vypisování průběhu je hráči zamezeno, aby nemohl měnit své zbraně a znovu tak může zaútočit až po ukončení předchozího zápasu. Pro výpis zde používáme IEnumerator writeNice(), který zajistí že vypisování má hezkou animaci.

Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky

Obrázek - animace výpisu textu

## Implementace Zvuků – Matěj Černý

Zvuky v naší aplikaci jsou řešeny pomocí objektů, které spravují přehrávání zvuků, nazývaných AudioManagery. Máme dva druhy těchto AudioManagerů, univerzální a unikátní. Oba tyto objekty fungují podobně až na pár výjimek. AudioManager má přiřazeno pole objektů zvuků, které mají definované různé atributy, jako například název, odchylku od hlasitosti a danou zvukovou stopu. AudioManager má právě jeden zdroj zvuku (AudioSource), do kterého se po každém zavolání metody play(zvuk) dosadí příslušná zvuková stopa a další argumenty (hlasitost, název atd). Proto se tak nedají v jednu chvíli překrývat dva zvuky ze stejného AudioManageru. Přehrávání hudby funguje na naprosto stejném principu jako přehrávání zvuků, s tím rozdílem, že se zvuková stopa přehrává neustále dokola.

Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky

Obrázek - práce se zvukem

Hlavním rozdílem mezi univerzálním a unikátním AudioManagerem je obsažení zvuků. Univerzální obsahuje zvuky, které se používají v UI vrstvě a které se vyskytují ve více scénách. Univerzální AudioManager se přenáší mezi scénami a je proto jen jeden v celém projektu. Také se používá k přehrávání hudby a zajišťuje tak, aby vždy hrála jen jedna hudba a nepřeskakovala při přepínání mezi jednotlivými scénami. Unikátní AudioManager je v každé scéně právě jeden, a zvuky, které spouští, jsou tak unikátní pro danou scénu. Také nedokáže spouštět hudbu. Univerzální AudioManager je potomkem unikátního, který rozšiřuje. Zajímavá je také synchronizace jednotlivých AudioManagerů a jejich hlasitosti, kterou můžeme měnit. Když totiž změníme hlasitost, změna se může promítnout pouze do AudioManagerů, které existují v momentální scéně. Proto máme metody PullSynchronizeVolume() a PushSynchronizeVolume(), které vždy při vytvoření nového unikátního AudioManageru (při přepínání mezi scénami) synchronizují hlasitost nově vytvořeného objektu s Univerzálním AudioManagerem. Zároveň tyto metody zajišťují při změně hlasitosti i změnu hlasitosti všech momentálně existujících AudioManagerů.

## Systém Ukládání – Matěj Černý, Václav Mašl

V Naší Aplikaci nemáme pevné ukládání dočasné (při vypnutí aplikace nezůstane nic uloženo). Dočasné ukládání za běhu aplikace využíváme především při přechodu mezi scénami. To zajištuje, aby při odchodu a následném návratu do místnosti byla místnost ve stejném stavu jako po odchodu. K vysvětlení toho, jak náš systém ukládání funguje je třeba nejdříve říct, že naše ukládání nefunguje na principu ukládání stavů ale spíše jednotlivých atributů a vlastností dané místnosti. Téměř veškeré objekty, se kterými uživatel, jakkoliv interaguje nejsou přímo staticky ve scéně. Při načtení scény se na objektu saver spustí script sceneStart a ten zajistí že se na scéně vytvoří všechny tyto potřebné objekty. Ty se inicializují z prefabů pomocí metody Instantiate(). Objekty k inicializeci získáme ze ScriptableObjektu dané místnosti, které zde používáme k dočastnému ukládání.

Obsah obrázku text, řada/pruh, Písmo, snímek obrazovky

Popis byl vytvořen automaticky

Obrázek - ukládání

U vytvořených objektů rovnou nastavíme jejich pozici, kterou jsme získali taktéž z našeho ScriptableObjekt úložiště. Díky tomu že tyto objekty nejsou staticky v místnosti, ale jejich vytvoření je závislé na datech z úložiště se nám tak ušetří spousta problémů, které bychom měli, kdybychom k ukládání přistupovali jako ke stavům místnosti.

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Popis byl vytvořen automaticky

Obrázek - SceneClass

Třída SceneClass nám skvěle ukazuje, jak jednoduše se teď dá k ukládání přistupovat. K přidání objektu stačí pouze zavolat metodu AddItem() s a jako argumenty dát daný item a jeho pozici. Jeho odebrání je stejně snadné. Díky tomu tak můžeme jednoduše a přehledně ukládat i pozice jednotlivých Objektů. Takže například když necháme v místnosti pod postelí a poté se do ní vrátíme nalezneme ho na stejném místě kde jsme ho nechali, a ne na místě kde je jeho výchozí pozice. Tyto metody můžeme volat z jakéhokoliv skriptu ze kterého potřebujeme a můžeme tak dočasně ukládat na základě libovolných událostí. Toto je ovšem jen Třída pro ulehčení práce s ukládáním samotné ukládání se probíhá v metodě write() kde se ArrayListy Třídy SceneClass uloží do Arraylistů ScriptableObjektu dané třídy. Metoda read() funguje naopak(Naplní SceneClass hodnotami ze ScriptableObjektu).Tyto metody jsou volány vždy jen při změně scény aby nedocházelo k neustálému zbytečnému zápisu do ScriptableObjektu ale pouze v případech kdy je zápis relevantní. Každá scéna má vlastní objekt saver který zajištuje inicializaci itemů a jejich ukládání.

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Popis byl vytvořen automaticky

Obrázek 42- write/read itemů

### ScriptableObject jako úložiště

ScriptableObject je datový kontejner, který můžete použít k uložení dat nezávisle na instancích třídy. Jedním z hlavních případů použití ScriptableObjects je snížení využití paměti projektu tím, že se vyhnete kopírování hodnot. Data, která uložíte z Editor Tools do ScriptableObjects jako aktivum, jsou zapsána na disk, a proto jsou mezi relacemi trvalá. ScriptableObject tak slouží skvěle pro naše účely vzhledem k tomu, že drží své hodnoty mezi relacemi, ale po ukončení aplikace se zárověň vrátí do výchozího stavu. Proto jsme se je rozhodli je upřednostnit před pevným ukládáním např. do. json nebo .csv souborů. Se ScriptableObjekty se také přívětivěji pracuje v unity Inspectoru. Tyto objekty v našem projektu ovšem nepoužíváme jen jako uložiště ale třeba i pro reprezentaci jednotlivých Itemů kdy mají ScriptableObjekty pevná neměnná data. Toho využíváme především v UI vrstvě v inventáři, jelikož ScriptaleObject narozdíl od Monobehavior nevyžaduje připojení na GameObject. Díky tomu s nimi můžeme v inventáři operovat nezávisle na jakémkoliv GameObjectu.

### Změna scény

Další část projektu, která byla řešena hned na začátku bylo měnění scén. Pro tento účel jsme využívaly knihovnu SceneManager, která práci velmi ulehčila.

V následujícím obrázku lze vidět, jak scéna vypadá v momentálním stavu hry.



Obrázek 43- třída changeScene

Na začátku tento script začínal jako obyčejný skript, který pouze změnil scénu.

Posléze jsme přidali delay při načtení a také automatické ukládání stavu místnosti při změně. Především ukládání stavu nám velmi pomohlo při práci se scénami.

Celková práce se scénami v Unity byla poměrně snadná a také lehce pochopitelná. Osobně jsme se rozhodli pro určení scén pomocí indexů, ale je možnost přepínat i pomocí názvu scény.

## Testování

Testování jsme prováděli tak, že jsme několika spolužákům dali přístup k demo buildu a nechali jsme je hrát. Následně s námi komunikovali, co ve hře našli za nedostatky a co by upravily.

Na testování sice nebylo příliš mnoho času, ale stejně přineslo cenné poznatky.

Jako jednu hlavní chybu, kterou bychom nejspíše bez testerů nenašli beru odcházení z miniher, ke kterým potřebujeme poznatek z jiné místnosti. Nás to osobně nenapadlo, protože jsme věděli kombinace a hesla, ale pro běžného hráče by se tímto stala hra nehratelnou.

Vzali jsme v potaz také několik poznatků ohledně vzhledu a dalších maličkostí.

Závěr

Na závěr bychom se chtěli vyjádřit k průběhu naší práce. Na začátku nám přišlo, že máme mnoho času, avšak tomu tak nebylo, jak jsme si časem uvědomili. Na začátku jsme pracovali více z pozvolna a ke konci jsme měli co dělat, abychom vše splnili do nadcházejícího sprintu. I přesto si myslíme, že jsme tomu dali vše, co v nás bylo, a že se nám hra nakonec vyvedla, i když ne v našem vysněném stavu.

Většina změn, které jsme se rozhodli v průběhu tvorby udělat byly výhradně kvůli nedostatku času. Našli se však i změny, na které jsme přistoupili kvůli větší logičnosti, nebo snazší implementaci.

Nic jsme však ve finálním projektu nevynechali, i když by mohli být všechny minihry více logické a náročné, jejich funkčnost ve hře je a funguje. Kdybychom se někdy k projektu měli vrátit, určitě bychom zvýšili úroveň všech miniher a logické návaznosti mezi nimi. Dále bychom více vyladili všechen pixel art, jelikož posledních pár místností bylo uděláno ve spěchu.

Celkově z projektu máme spíše pozitivní pocit. Všichni jsme se z práce na něm přiučili mnoho věcí, které budeme schopni uplatnit v nějakých budoucích pracích, či projektech. Snad největším přínosem pro nás byla práce a komunikace v týmu, naučili jsme se, že bez komunikace to zkrátka nejde. Na svoji práci jsme pyšní a v budoucnu můžeme odkazovat na všechny zkušenosti, které jsme zde nabyli.

Seznam obrázků

[Obrázek 1- ScriptableItem class 17](#_Toc162338478)

[Obrázek 2- PickupItem 18](#_Toc162338479)

[Obrázek 3 - Recipe class 19](#_Toc162338480)

[Obrázek 4- metoda create 19](#_Toc162338481)

[Obrázek 5- ukázka PauseMenu 20](#_Toc162338482)

[Obrázek 6- metoda animatedText 21](#_Toc162338483)

[Obrázek 7- metoda setNextHint 21](#_Toc162338484)

[Obrázek 8- metoda dialogEnded 22](#_Toc162338485)

[Obrázek 9- metoda checkInventory 22](#_Toc162338486)

[Obrázek 10- inspirace 23](#_Toc162338487)

[Obrázek 11- ukázka prostředí Aseprite 24](#_Toc162338488)

[Obrázek 12- návrh místnosti 25](#_Toc162338489)

[Obrázek 13- místnost 2 26](#_Toc162338490)

[Obrázek 14- ukázka highlight 26](#_Toc162338491)

[Obrázek 15- item torch 27](#_Toc162338492)

[Obrázek 16- inspirace minihry 28](#_Toc162338493)

[Obrázek 17- návrh minihry 28](#_Toc162338494)

[Obrázek 18- minihra rozbití zámku 29](#_Toc162338495)

[Obrázek 19- tečka 29](#_Toc162338496)

[Obrázek 20- minihra runy 30](#_Toc162338497)

[Obrázek 21- title screen 31](#_Toc162338498)

[Obrázek 22- návrh tlačítka exit 31](#_Toc162338499)

[Obrázek 23- ovládací šipka 32](#_Toc162338500)

[Obrázek 24- soundbar 32](#_Toc162338501)

[Obrázek 25- inventář 33](#_Toc162338502)

[Obrázek 26- bossfight 34](#_Toc162338503)

[Obrázek 27- Audacity 35](#_Toc162338504)

[Obrázek 28- MusicFX 36](#_Toc162338505)

[Obrázek 29- spouštění miniher 36](#_Toc162338506)

[Obrázek 30- ukázka Zlomení zámku 37](#_Toc162338507)

[Obrázek 31- ukázka dotMovement 37](#_Toc162338508)

[Obrázek 32- ukázka minihry Runy 38](#_Toc162338509)

[Obrázek 33- nastavení spritů 38](#_Toc162338510)

[Obrázek 34- ukázka změni spritu a ověření 39](#_Toc162338511)

[Obrázek 35- ukázka vložení spritů v Unity GUI 39](#_Toc162338512)

[Obrázek 36- ukázka Souhvězdí 40](#_Toc162338513)

[Obrázek 37- ukázka kontroly hesla 40](#_Toc162338514)

[Obrázek 38- animace výpisu textu 41](#_Toc162338515)

[Obrázek 39- práce se zvukem 42](#_Toc162338516)

[Obrázek 40- ukládání 43](#_Toc162338517)

[Obrázek 41- SceneClass 43](#_Toc162338518)

[Obrázek 42- write/read itemů 44](#_Toc162338519)

[Obrázek 43- třída changeScene 45](#_Toc162338520)