Střední průmyslová škola elektrotechniky a informatiky, Ostrava

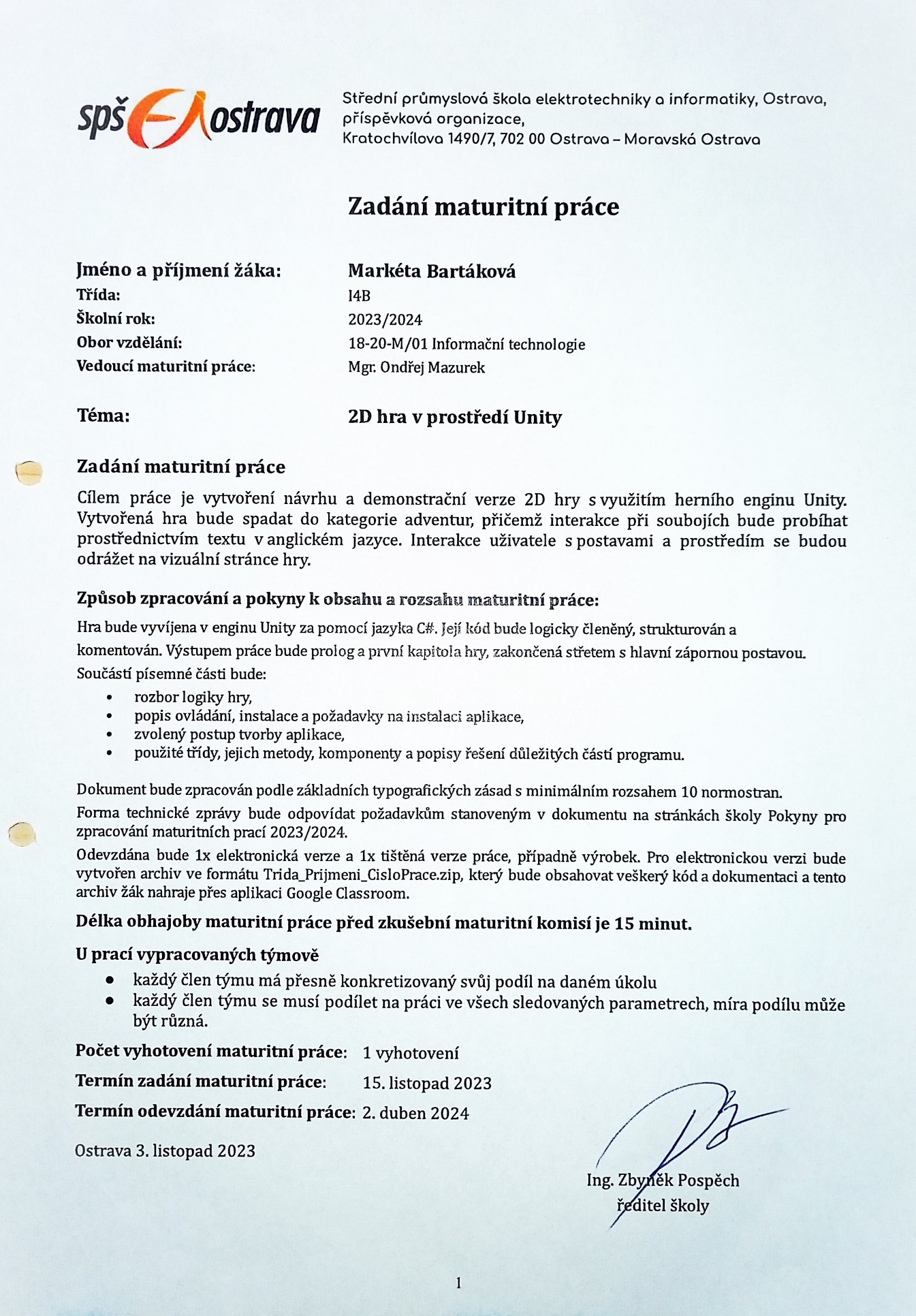
**25. 2D hra v prostředí Unity**  
Maturitní práce

Autor práce: Markéta Bartáková

Vedoucí práce: Mgr. Ondřej Mazurek

Třída: I4B

Školní rok: 2023/2024

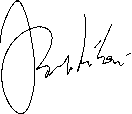


Prohlašuji, že předložená práce je mým původním dílem, které jsem vypracoval samostatně.

Veškerou literaturu a další zdroje, z nichž jsem při zpracování čerpal, v práci řádně cituji

a jsou uvedeny v seznamu použité literatury a zdrojů informací.

V Ludgeřovicích 2. 4. 2023 podpis:



Licenční ujednání

Ve smyslu §60 autorského zákona č. 121/2000 Sb. poskytuji, Střední průmyslové škole

elektrotechniky a informatiky, Ostrava, příspěvkové organizaci, Kratochvílova 1490/7,

702 00 Ostrava, bezplatně oprávnění k výkonu práva (licenci) ke školnímu dílu (maturitní

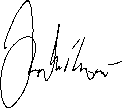
práci) užít v rozsahu a způsoby uvedenými v §12 až 23 autorského zákona.

Souhlasím / Nesouhlasím se zveřejněním díla v rámci školní počítačové sítě pro potřeby

studentů a zaměstnanců školy a pro potřeby výuky v souladu s §35(3) autorského zákona.

Souhlasím / Nesouhlasím s použitím práce k propagaci školy.

V Ludgeřovicích 2. 4. 2023 podpis:



Anotace

Tato práce popisuje postup vývoje počítačové hry. Od zrodu myšlenky až po finální testování. Text zahrnuje popis jednotlivých komponent a kódů hry, její příběh, motivace, použité technologie, ale i popis postupného odlaďování problémů a nastínění vize budoucího vývoje. Součástí je i návod pro uživatele, jak hru zapnout a jak se v ní orientovat, a výpis aktuálních problémů, které nebylo možno vyřešit, jejich možné příčiny, a vhodné budoucí optimalizace.

Poděkování

Chtěla bych poděkovat svému vedoucímu práce, Mgr. Ondřeji Mazurkovi, za umožnění uskutečnění tohoto projektu a za jeho stálou ochotu a podporu. Také bych chtěla poděkovat Nele Březinové, která mě zachránila v největší krizi, Ing. Ivu Zapletalovi za praktické rady a připomínky, Mgr. Vlastně Kubinové za poskytnutí podstatného základu v programování. Jako poslednímu bych chtěla poděkovat svému bratrovi, Marku Bartákovi, který byl nucen poslouchat hodiny mých proslovů, avšak pomáhal mi přijít na ta správná řešení a nápady. Bez těchto lidí by projekt neměl šanci ani vzniknou, ani být v takové podobě, jaké je dnes. Děkuji vám všem, bez vás bych to nedokázala.

Obsah

[Anotace 4](#_Toc162314955)

[Poděkování 5](#_Toc162314956)

[1 Úvod 9](#_Toc162314957)

[2 Co je to počítačová hra 10](#_Toc162314958)

[2.1 Videoherní průmysl 10](#_Toc162314959)

[2.2 Žánry počítačových her 10](#_Toc162314960)

[2.3 Co je to herní engine 11](#_Toc162314961)

[2.4 Nejpoužívanější enginy dnes 11](#_Toc162314962)

[3 Použité technologie a nástroje 12](#_Toc162314963)

[3.1 Unity 12](#_Toc162314964)

[3.2 C# 12](#_Toc162314965)

[3.3 Rider 12](#_Toc162314966)

[3.4 GitHub 12](#_Toc162314967)

[3.5 GitHub Desktop 12](#_Toc162314968)

[4 Počátky vývoje 13](#_Toc162314969)

[4.1 Motivace 13](#_Toc162314970)

[4.2 Původní téma 13](#_Toc162314971)

[4.3 Hlavní mechaniky 13](#_Toc162314972)

[4.4 Ujasnění prvků hry 13](#_Toc162314973)

[5 Vývoj prologu 14](#_Toc162314974)

[5.1 Cíl 14](#_Toc162314975)

[5.2 Hlavní postava 14](#_Toc162314976)

[5.3 Překopání celého projektu 14](#_Toc162314977)

[5.4 Nová vize 14](#_Toc162314978)

[5.5 Nový příběh 14](#_Toc162314979)

[5.5.1 Mládí 14](#_Toc162314980)

[5.5.2 Dospívání 15](#_Toc162314981)

[5.5.3 Dospělost 16](#_Toc162314982)

[5.6 Příběh prologu 16](#_Toc162314983)

[5.7 Závěr práce na prologu 17](#_Toc162314984)

[6 Kódová stránka tutoriálu 17](#_Toc162314985)

[6.1 Ovládání postavy 17](#_Toc162314986)

[6.2 CharacterController2D 17](#_Toc162314987)

[6.2.1 LoadInputs() 18](#_Toc162314988)

[6.3 MovingAround() 18](#_Toc162314989)

[6.3.2 FixedUpdate() 19](#_Toc162314990)

[6.4 Ovládání skoku 19](#_Toc162314991)

[6.5 CharacterBehavior 20](#_Toc162314992)

[6.6 WritingGameplay 20](#_Toc162314993)

[6.6.1 BuffWord() 20](#_Toc162314994)

[6.6.2 Update() 21](#_Toc162314995)

[6.7 Následování textového pole postavy 22](#_Toc162314996)

[6.8 Chování kamery 23](#_Toc162314997)

[6.8.1 MoveCam() 23](#_Toc162314998)

[6.8.2 ZoomCam() 24](#_Toc162314999)

[6.9 Interakce s objekty scény 24](#_Toc162315000)

[6.9.1 Vypisování textu 25](#_Toc162315001)

[6.9.2 Interakce 25](#_Toc162315002)

[7 Entre level 28](#_Toc162315003)

[7.1.1 Coroutines 29](#_Toc162315004)

[7.1.2 IEnumerator PrintSentences() 29](#_Toc162315005)

[8 Kódová stránka prologu 30](#_Toc162315006)

[8.1 Přecházení na další levely 31](#_Toc162315007)

[9 Levely uvnitř domu 32](#_Toc162315008)

[10 Boss Level 33](#_Toc162315009)

[10.1 Systém boje 33](#_Toc162315010)

[10.2 Práce s bojovými slovy 34](#_Toc162315011)

[10.3 Řízení pořadí vypisování 34](#_Toc162315012)

[10.4 Váha slov 35](#_Toc162315013)

[10.5 Třída WriteIntoBubble 35](#_Toc162315014)

[10.5.1 V této části kódu jsou nejpodstatnější: 36](#_Toc162315015)

[10.6 Ukázky ze souboje 37](#_Toc162315016)

[10.7 Různé možnosti konce 38](#_Toc162315017)

[11 Další prvky hry 38](#_Toc162315018)

[11.1 Main menu 38](#_Toc162315019)

[11.2 Pause menu 38](#_Toc162315020)

[12 Potíže při vývoji 38](#_Toc162315021)

[12.1 Nefungující rigování 38](#_Toc162315022)

[12.2 Nelze zvolit položky Pause menu v levelech 5 a 6 38](#_Toc162315023)

[12.3 Špatný výpis řetězců pro interakci 39](#_Toc162315024)

[13 Nevyužité nápady 39](#_Toc162315025)

[13.1 Systém hydratace 39](#_Toc162315026)

[13.2 Papoušek jako mazlíček 39](#_Toc162315027)

[13.3 Možnost používání mobilu 39](#_Toc162315028)

[13.4 Původní systém psaní 40](#_Toc162315029)

[14 Budoucí vývoj 40](#_Toc162315030)

[14.1 Lepší interakce 40](#_Toc162315031)

[14.2 Sbírání knih s příběhem 40](#_Toc162315032)

[14.3 Možnost uložení a načtení hry 40](#_Toc162315033)

[14.4 Kosmetické úpravy 40](#_Toc162315034)

[Citovaná literatura 43](#_Toc162315035)

# Úvod

Cílem práce bylo vytvoření návrhu a demonstrační verze 2D hry s využitím herního enginu Unity a jazyku C#. Vytvořená hra má spadat do kategorie adventur, přičemž interakce s prostředím probíhá prostřednictvím textu v anglickém jazyce. Hra má obsahovat prolog, první kapitolu a je zakončena střetem s hlavní zápornou postavou.

Vždy jsem byla kreativní jedinec, kterého lákalo i programování. Tak jsme si jednoho dne s kamarádkou řekly, proč nezkusit společně vytvořit hru, tím že je obě rády hrajeme a máme podobný vkus. Chtěly jsme si odzkoušet dovednost, kterou bychom chtěly v budoucnu využít při kariéře. Kamarádka přišla s nápadem na hru, rozdělily jsme si role – já budu programovat, ona bude mít na starost grafiky. A daly jsme se do společného vývoje.

# Co je to počítačová hra

Jedná se o hru realizovanou prostřednictvím počítače. Jejím cílem je zprostředkovat nám virtuální realitu. Je zvláštní médiem pro zpracování příběhu, či prostou relaxaci. Příběh díky spojení vizuálu, zvuku ale i imerze tudíž umí mít kolikrát mnohonásobně vyšší dopad.

Nejedná se však o nikoliv staré médium. První počítačové hry datují již do 60. let 20. století. Prvotiny jako [OXO](https://en.wikipedia.org/wiki/OXO_(video_game)), [Tennis for Two](https://en.wikipedia.org/wiki/Tennis_for_Two) a [Spacewar!](https://cs.wikipedia.org/wiki/Spacewar!) zahájily vzrůst nadšení pro počítačové hry a byly zrodem nového průmyslu. [1]

## Videoherní průmysl

V této době patří k poměrně podstatné části zábavního průmyslu a zaměstnává velkou variaci lidí, od grafiky, po programátory, zvukaře, scénáristy apod. V posledních letech se dostal obrovského vzrůstu, obzvláště za doby pandemie, kdy byly tisíce lidí uvězněny doma. Světové roční tržby herního průmyslu činí zhruba 200 miliard USD (4,6 bil. korun). To odpovídá více než dvojnásobku výdajů českého státního rozpočtu pro rok 2023. V současnosti hraje pravidelně počítačové hry po celém světě podle odhadů 3,4 mld. lidí (z celkových 8 mld.) [2].

## Žánry počítačových her

Jde o klasifikaci videohry, která je založena na druhu interakce, vizuální a příběhové stránky. Mezi žánry kolikrát bývá tenká čára, a tudíž se často i míchají mezi sebou. Každý žánr má však své charakteristické prvky: [3]

**Akce**

* Podstatná určitá míra násilí a adrenalinu
* Po hráči jsou vyžadovány zejména rychlé reflexy či precizní míření
* [Assasin’s Creed](https://store.steampowered.com/app/812140/Assassins_Creed_Odyssey/)
* [Tomb Rider](https://store.steampowered.com/app/203160/Tomb_Raider/)
* [Sekiro: Shadows Die Twice](https://store.steampowered.com/app/814380/Sekiro_Shadows_Die_Twice__GOTY_Edition/)

**Adventury**

* Základem je zajímavý příběh, který hráč postupně prochází
* 2D adventury se zakládají na osobitých grafikách
* Lze kombinovat s různými logickými úlohami
* [Hollow Knight](https://store.steampowered.com/app/367520/Hollow_Knight/)
* [Limbo](https://store.steampowered.com/app/48000/LIMBO/)
* [Terraria](https://store.steampowered.com/app/105600/Terraria/)

**Strategie**

* Základem logické uvažování
* Jeden z mála žánrů, který zůstává čistě pro PC
* Budování, karetní hry apod.
* [Hearthstone](https://hearthstone.blizzard.com/en-us)
* [Surving Mars](https://store.steampowered.com/app/464920/Surviving_Mars/)
* [Hearts Of Iron](https://store.steampowered.com/app/25890/Hearts_of_Iron_III/)

**RPG**

* Většinou volný otevřený svět
* Hry na hrdiny
* Rozvinutý příběh, vývoj postavy
* [Baldur’s Gate](https://store.steampowered.com/app/1086940/Baldurs_Gate_3/)
* [The Witcher](https://store.steampowered.com/app/292030/Zaklna_3_Divok_hon/)
* [Final Fantasy](https://store.steampowered.com/app/39210/FINAL_FANTASY_XIV_Online/)

## Co je to herní engine

Herní engine je software sloužící k vývoji videoher. Termín herní engine se poprvé začal objevovat v polovině devadesátých let ve spojitosti s hrami typu FPS. Dobrým příkladem byla hra Doom od Id Software. Tato hra měla rozumným způsobem oddělené jádro (vykreslování 3d grafiky, detekce kolizí, audio systém…) a vlastní náplň hry (prostředí, zvuky, pravidla hry...). Toto rozdělení se ukázalo být velmi výhodné, když Id Software začal toto jádro licencovat dalším firmám. Ty se nemusely starat o nízko úrovňové aspekty hry, a stačilo jim jenom vytvořit vlastní prostředí a vlastní pravidla hry. To velmi zrychlilo a zlevnilo vývoj hry. [4]

## Nejpoužívanější enginy dnes

|  |  |
| --- | --- |
| 1. | Unreal Engine |
| 2. | Unity |
| 3. | Godot |
| 4. | Amazon Game Engines (Lumberyard, Open3D) |
| 5. | CryEngine |

Tabulka 1: Statistika engine za rok 2023 [5]

# Použité technologie a nástroje

## Unity

Unity je multiplatformní herní engine vyvinutý společností Unity Technologies. Byl použit pro vývoj her pro PC, konzole, mobily a web. První verze podporovala pouze OS X a byla představena na celosvětové konferenci Applu v roce 2005. Od té doby byl rozvinut o více než patnáct dalších platforem.

Unity poskytuje možnosti vývoje pro 2D i 3D hry libovolného žánru a zaměření. Kromě grafického prostředí pro tvorbu, podporuje také tvorbu skriptů v jazyce C#. [6]

## C#

C# je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework, později schválený standardizačními komisemi ECMA (ECMA-334) a ISO (ISO/IEC 23270). Microsoft založil C# na jazycích C++ a Java (a je tedy nepřímým potomkem jazyka C, ze kterého čerpá syntaxi).

C# lze využít k tvorbě databázových programů, webových aplikací a stránek, webových služeb, formulářových aplikací ve Windows, softwaru pro mobilní zařízení (PDA a mobilní telefony) atd. [7]

## Rider

JetBrains Rider je multiplatformní .NET IDE založené na platformě IntelliJ a ReSharperu.

Rider poskytuje sílu pro produktivní vývoj široké škály aplikací, včetně .NET desktopových aplikací, služeb a knihoven, her v Unity a Unreal Engine, aplikací Xamarin, ASP.NET a ASP.NET Core webových aplikací a dalších. Na platformách Windows, macOS a Linux. [8]

## GitHub

GitHub je webová služba podporující vývoj softwaru za pomoci verzovacího nástroje Git. GitHub nabízí bezplatný Webhosting pro open source projekty. Od 7. ledna 2019 je možné ukládat bezplatně i soukromé repositáře (dříve po zaplacení měsíčního poplatku). Projekt byl spuštěn v roce 2008, zakladatelé byli Tom Preston-Werner, Chris Wanstrath a PJ Hyett. [9]

## GitHub Desktop

GitHub Desktop je zdarma dostupná open source aplikace, která vám pomáhá pracovat soubory hostovanými na GitHubu nebo jiných službách pro hostování Gitu. [10]

# Počátky vývoje

## Motivace

Hlavní motivací toho projektu bylo odzkoušet si budoucí povolání. Jak já, tak kamarádka jsme vždy snily o práci ve vývoji her. Tento projekt měl otestovat a demonstrovat naše schopnosti, mé v programování, její v grafice.

## Původní téma

Hlavním tématem a inspirací byla dnešní doba a s ní spjata váha slov. Již od nepaměti patřila k lidské komunikaci slova. Avšak ne všichni ví, jak s nimi správně pracovat. Někteří si neuvědomují jejich váhu, jiní je zneužívají. Našim posláním bylo poukázat na morální stránku slov a jaký mohou mít efekt na člověka a to, jak člověk v čase konfrontace často degraduje na používání slovníku protější osoby.

## Hlavní mechaniky

Za hlavní mechaniky hry jsme se rozhodly použít právě slova. Naše postava avšak nemá moc na výběr, jaká slova bude používat. Pouze při konfrontaci otce. Tato mechanika poukazuje na to, jak kolikrát ani nepřemýšlíme nad slovy a nesnažíme se mezi nimi nijak volit. Hlavní inspirací těchto mechanik nám byla hra [Textorcist](https://store.steampowered.com/app/940680/The_Textorcist_The_Story_of_Ray_Bibbia/), která k interakci se světem používá právě slova.

## Ujasnění prvků hry

Již ze začátku jsme věděly, že chceme do hru vést především textem. Avšak pohyb jsme se rozhodly nechat klasicky na ovládání šipkami a WASD.

Součástí hry původně měly být i různé minihry a integrovaný mobil. Na mobilu by se ony hry právě hrály a šlo by jím i fotit.

Mezi možné minihry patřily: floppy bird, komentování příspěvků na sítích, učení papouška trikům

# Vývoj prologu

## Cíl

Hlavním cílem prologu je naučit hráče mechaniky (pohyb a interakce) a jejich prolínání. Zároveň nastiňuje celou situaci postavy a jejího světa.

## Hlavní postava

Postava, kterou kamarádka pojmenovala Milan (nemá spojitost s žádnou existující osobou), je věčně zavřený doma a účastní se různých diskusí na online fórech. Díky tomu ale ztrácí schopnost ovládat se, rychle se naštve, musí se ihned ke všemu vyjádřit a jeho život se otáčí kolem života ostatních. Svou zlost na svět si vylévá právě na daných fórech. Tím degraduje jeho mentalita, a jednoho dne se dostane do nepříjemné konfrontace, kdy se projeví tato degradace mysli. Podle jména postavy byla původně i pojmenována hra.

## Překopání celého projektu

Při vývoji projektu došlo k mnoha neshodám s nyní již bývalou kamarádkou a spoluautorkou. Tyto neshody vedly k ukončení naší spolupráce a zbyla jsem na projekt sama. Původní nápad se mi stále zamlouval, byť ironicky poukazoval na celou absurditu této situace. Ale už jen kvůli tomu, že původní mechaniky by mi neustále připomínaly danou situaci, ale i kvůli tomu, že jsem nevěděla jak a kam to vést, jsem se rozhodla změnit přístup a téma.

Jelikož se ztrátou jednoho autora ztratila hra z části i svou identitu, rozhodla jsem se projekt přejmenovat místo **Milan** na **M1?4N**. Jak název, tak finální hra tudíž obsahují jak staré prvky spolupráce, tak prvky práce vlastní.

## Nová vize

S novou verzí hry a příběhu jsem chtěla stále zachovat jádro, a to degradaci mysli vlivem vnějších faktorů. V původní práci se jednalo o slova. Já jsem se je však rozvinula o vizuály a přidala mezi ně kontrast. To, co je vyobrazeno, je kolikrát v rozporu se slovy. Toto značí diskonekci mozku od reality. Nová verze hry tudíž pracuje s tématem traumatu. Tato volba byla inspirována mou oblíbenou hrou, [Fran Bow](https://store.steampowered.com/app/362680/Fran_Bow/), na níž jde vidět postupná gradace mysli a distance od reality.

## Nový příběh

S novou vizí přišel nový příběh, tentokrát o návratu traumatizovaného jedince do prostředí traumatu:

### Mládí

Milan byl vždy zvláštní dítě. Neuměl se moc regulovat, nikdo jej to nenaučil. Měl však jednoho přítele. Byl sice jenom jeden, ale ti dva si byli bližší než atomy kovů. Ale pouze do určitého incidentu.

Tak jako normální den si Milan hrál s kamarádem u útesu nad vodou. Rádi pozorovali a chytali ryby. Kamarád jich měl vždy více, ale Milanovi to nevadilo. Toho dne společně ulovili obrovskou rybu, největší, jakou kdy chytili. Oběma udělala neskutečnou radost. Chtěli se s ní jít pochlubit, ale Milan si uvědomoval, že takto by ryba zahynula. Kamarád si nechtěl dát říct. Stál si za svým. O rybu se tedy začali přetahovat. Krok, co krok, byli blíže útesům, zastavili se až na jeho samostatném kraji. „Hoď ji zpátky,“ Naříkal Milan. Viděl, že ryba nemá už moc času. „Až ji ukážu,“ Řekl jeho kamarád a už se chtěl rozeběhnout směrem domů, když v tu Milanovi něco přepnulo v hlavě. Do kamaráda strčil. Neuvědomil si, co se stalo. V uších mu řinčel pouze křik. Těžká rána. Ticho. Milanovi ztuhla všechna krev v těle. Co jsem to udělal, proč jsem to udělal, míjelo se mu hlavou. Z krku nedokázal vydrat jedinou hlásku. Vykročil roztřesenou nohou a nejistými kroky se dostal až na pokraj útesů. Polkl. Sklonil zrak pod útesy. Všude červeň. Čerstvý pomalu stékal z kamenné postele, na které leželo malé tělo. Milan k němu sklopil oči. Střetli se pohledem. Milana polil chlad. Nedokázal se odtrhnout, dokud oči neztratily svůj lesk. Marně se s ním snažili třást vystrašení rodiče. Od té doby se změnil Milanův pohled na svět.

### Dospívání

Od toho incidentu nebylo nic jako před tím. Nejdříve se Milan bránil, že to byla pouze nehoda. Že kamarádovi se v rukou cukla ryba a on díky ní zakopl a spadl z útesu. Ohradili se proti němu ale všichni sousedé, i když tu situaci nikdo neviděl. Všichni jej nazývali lhářem, ale odmítali věřit tomu, že by teprve sedmi-leté dítě bylo schopno někoho zabít. Na vesnici nastaly spory takového rozměru, že musela být zapojena i policie. Případ prošetřili, avšak nemohli vyhodnotit jinak než jako nešťastnou náhodu. Nikdo tomu nevěřil. Ani Milanovi vlastní rodiče. „Jak jsi nám mohl něco takového udělat!? Kdo tě učil takovým věcem!?“ Tato slova se linula zdmi každý den, stejně ustavičně jako nekončící modřiny na Milanově těle.

Nejdříve přišla o práci matka. O celém incidentu se dozvěděla její kolegyně a nahlásila to jejich nadřízenému. Ten matku bezpodmínečně propustil, bez jakýchkoliv otázek. Stejným následkům podlehl i otec, avšak ten si vyprosil možnost dále pracovat, dokud si nenajdou nové místo k žití. Čekal jej však posun na stále nižší pozice, dokud neuměli z peněz vyžít. Nebýt Milanovy pověsti na vesnici, jistě by jej nutili pracovat bez přestávky. Přece jenom byli přesvědčeni, že on za to mohl. Avšak práce ani peníze nebyly, a tak se rodina jednoho dne sbalila a přestěhovali se na novou vesnici poblíž velkého města. Vypadalo to, že vše už bude lepší. Alespoň na chvíli.

Poklidný život nevydržel dlouho. Otec si sice našel úžasně placenou práci, ale matku nikdo nechtěl zaměstnat. Její bývalá společnost, ve které dělala, brzy po jejím odchodu zkrachovala. A tak se ji nikdo nechtěl ujmout, již čistě z principu. Snažila se to všemožně obejít: ucházela se o práci v kavárně, restauraci, knihovně, nemocnic, ale stále nic. A tak se rozhodla zůstat doma. Milana umístili do školy nedaleko domu. Nejednalo se o vůbec náročnou školu, ale Milan měl problém jak se studiem, tak zapadnout do kolektivu. Nikdo se s ním nechtěl bavit, prý vyzařoval divnou auru, štěbetaly si o něm děti. A jakmile jednoho dne zjistily pravdu, nenazvaly jej jinak než slovy: monstrum, vrah a další, více brutální výrazy. Milan se ze školy vracel často dobitý, učitelé dělali, že to nevidí, rodiče to ještě vychvalovali: „To ti patří za to, jak jsi nám zničil životy.“

Milan školu nikdy nedokončil. Několikrát musel opakovat ročník, a nakonec jej v sedmnácti letech vyřadili. Trávil tak většinu času doma. Věděl, že nemá šanci najít si práci, a tak celé dny proseděl za počítačem a projížděl různá fóra. Sem tam z domu zmizel na pár hodin. Jeho rodiče vždy doufali, že se tentokrát už nevrátí. Avšak marně. Vždy se vrátil těsně za západu slunce s novým kamínkem v kapse.

### Dospělost

Jednou se ale domů vrátil dříve, než měl. Našel tam tak otce s nějakou cizí paní. Jen co ji spatřil, chtěl běžet pryč. Věděl, že objevil něco, co neměl. Než však stihl utéct, popadl jej otec za vlasy, začal na něj nadávat, mlátit jej a vyhrožovat: „Jestli se o tom dozví tvá matka, tak si mě nepřej!“ A tak Milan držel celou dobu jazyk za zuby. Dařilo se mu to rok, než je přistihla sama matka. Tu noc nikdo v sousedství nespal. Několikrát se na prahu dveří objevila policie, avšak nikdy nezasáhla. Ani když se doprostřed boje omylem dostal Milan. Chtěl jít ven, utéct tady tomu všemu. Místo toho se ale připletl mezi otce a matku a od obou obdržel četná zranění nožem. Přežil jen díky jednoho policisty, kterému to nedalo a vpadl na scénu.

Brzy na to se rozešli. Dilema však bylo, kdo se ujme Milana. Ne, že by odsoudili za ublížení na zdraví, vůbec. Všichni nad tím přivřeli oči. Nakonec však Milan připadl matce, jelikož matka se kvůli zranění utržených při té hádce o sebe už nemohla sama starat. A tak se spolu odstěhovali zpět na vesnici, kde Milan vyrůstal. Jeden by čekal, že se mu pomalu začnou vracet vzpomínky na dětství a vše co se stalo. Avšak ony jej nikdy neopustily. Milan na tom místě incidentu trávil většinu svých dnů. Tam, kde kdysi byly ty vyčuhující kameny, již byla voda, která vše smyla. Ale i tak Milan před očima stále vidí to mladé, hynoucí tělo.

Nikdo z vesnice nevěděl, že je Milan zpět. Věděli pouze o jeho matce a ta jej nikdy nezmínila. Na otázky typu: „A co tvůj syn?“ vždy odpověděla suše slovy: „Nemám syna.“ Nikdo se dál nedoptával, většina žila v přesvědčení, že se jej jednoduše vzali. Proto byli tak zaskočeni, když jednoho dne potkali jeho otce a ptal se, kde je. Sousedi se polekaně ptali, proč a že ani nevěděli, že je Milan zpátky. Otec pár z nich dovedl k jejich původnímu domu. Vyrvané dveře, rozbitá okna, krev. Krev pokryla téměř každý povrch. Uprostřed obývacího pokoje visel zdroj této rudi: matka, rozpitvaná jako prase na jatkách, místnost ovinuta jejími orgány. Jen co to sousedé uviděli, vyletěli ze dveří. Ihned volali policii. Ti později Milana našli. Nebyli si ale jistí, jestli toto byla právě Milanova práce. Jeho otec si tím byl ale jist.

Policie ustavičně pracovala na vyřešení této vraždy. Ale jediné, k čemu došli, bylo to, že toto Milanova práce nebyla. I když tomu nahrával incident z minulosti, nedokázali jej s tím spojit. Ale ani nedokázali najít pravého viníka. Milan byl přesvědčen, že to byl jeho otec. Ale policie jej neposlouchala, už i jen kvůli tomu, že byl hlavní podezřelý. A tak bez prokázaného viníka byl proces ukončen a Milan se mohl navrátit domů.

## Příběh prologu

V prologu příběh volně navazuje na daný příběh. Milan se vrací po nějakém čase zpět do rodného místa. Zde si však ale začne uvědomovat minulost a její tíhu. Příběh prologu se spíš naklání k pohledu do jeho dětství a ke vzpomínkám na jeho přítele.

## Závěr práce na prologu

Výsledkem práce na prologu je plně sestavená scéna rodného místa a prohloubení schopnosti hráče ovládat mechaniky hry. Při vývoji jsem si ale všimla jistých nedostatků (například nevysvětlení způsobu pohybu a interakcí) a tak jsem se rozhodla vytvořil **Tutorial level**, který se přehraje hned jako první po spuštění hry. Tímto se minimalizuje zmatení hráče z mechanik a může svou pozornost věnovat prostředí hry a příběhu.

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

Popis byl vytvořen automaticky

Obrázek 1: Ukázka Prolog levelu ve vývojovém prostředí Unity

# Kódová stránka tutoriálu

V tutoriálu jde především naučit postupně hráče orientovat se v daném herním prostředí. Nejprve je mu tedy představen pohyb a po něm interakce s objekty.

## Ovládání postavy

Na hlavní postavě je nastavený Animator a Collider. Animator slouží ke správě animací, Collider řeší kolize. Obě tato rozhraní si řeší Unity editor. Zbytek akcí řeším přes třídy CharacterController2D a CHaracterBehavior.

## CharacterController2D

Tato třída řeší pohyb postavy a její chování při skocích.

### LoadInputs()

Ve funkci LoadInputs() načítá požadavky na pohyb od uživatele. Ve hře je možno se pohybovat dvěma způsoby: šipky nebo přes shift + (WAD). Tato funkce načte daný pohyb do jedné proměnné, tudíž nemusíme zvlášť řešit jakým způsob se pohybuje. Tímto zjistíme, jestli a kam se pohybujeme.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 1: metoda LoadInputs() třídy CharacterController2D

#### Rozdíl mezi GetKey(), GetKeyDown(), GetKeyUp()

Jedná se o metody třídy Input spravovanou Unity. Všechny funkce vrací bool hodnotu.

GetKey() – vrací true po celou dobu trvání stisku klávesy  
GetKeyDown() – vrací true pouze při první instanci stisku  
GetKeyUp() – vrací true při uvolnění klávesy

Ve funkci využívám GetKey() v případě, že zjišťuji jestli hráč stále drží danou klávesu.  
GetKeyDown() využívám pro případ triggeru – po stisku se požadovaný kód provede pouze jednou, nikoli neustále dokola jako v případě GetKey() (ta i v případě krátkého stisknu se spustí minimálně 4x).  
GetKeyUp() používám v případě, kdy zjišťuji jestli došlo k přerušení stisknu. Nikoli však k tomu, jestli hráč stále nedrží danou klávesu (k tomuto by sloužilo !GetKey()).

## MovingAround()

Funkce invokuje pohyb postavy dle inicializace pohybu načtenou z funkce LoadInputs().

#### Řešení otáčení postavy vlevo/vpravo

Obsah obrázku text, Písmo, snímek obrazovky

Popis byl vytvořen automaticky

Kód 2: Část metody Update() – ukázka otáčení

Vytvoříme nový 3D vektor, který nám překlápí rotaci postavy po ose Y. Hodnotu tohoto vektoru ukládáme do proměnné \_moveDir, kterou později využijeme ve funkci FixedUpdate().

### FixedUpdate()

FixedUpdate se obvykle používá pro výpočty fyziky, protože má stejnou frekvenci jako fyzikální systém: výchozími nastaveními se provádí každých 0,02 sekundy (50krát za sekundu). [11]

## Ovládání skoku

Obsah obrázku text, snímek obrazovky

Popis byl vytvořen automaticky

Kód 3: Část metody Update() – řešení jednoduchého skoku

Nejdříve se ověří, jestli se postava dotýká země (ta je nastavená pomocí Layers). Pokud postava doskákala a je není schopna znovu skočit (tato situace nastává při neuvolnění klávesy skok) tak daná část programu končí. Jinak se spustí skok. Při jednoduchém stisku skočíme velikostí uvedenou externě v editoru.

Pokud však chceme skákat po dobu držení klávesy, aplikuje se níže uvedený kód.

Obsah obrázku text, snímek obrazovky, software

Popis byl vytvořen automaticky

Kód 4: Část metody Update() – řešení drženého skoku

Při skákáni se nám odečítá čas, jak dlouho je ještě možno skákat. Tento čas je uveden externě v editoru. Po vypršení času přestáváme skákat.

## CharacterBehavior

Tento skript ovládá řídí animace postavy. Informace o řízení získává z CharacterControler2D.

Obsah obrázku text, snímek obrazovky

Popis byl vytvořen automaticky

Kód 5: Metoda AnimationGoIdleJump() třídy CharacterBehavior

Přehrání animace skoku je řešené po pomocí Triggeru – animace se přehraje pouze jednou, po skočení se automaticky vypne. Tato animace se i přeruší, jakmile se dotkneme země a tím přestaneme skákat.  
Animace pohybu jsou však řešeny přes bool hodnoty – animaci chceme přehrávat stále, dokud se nehýbeme.

## WritingGameplay

Tato třída pracuje se zadáváním textu hráčem.  
Je v ní určeno, kam a jak se má vypisovat input hráče.

### BuffWord()

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 6: Metoda BuffWord() třídy WritingGameplay

Uživatel může zkusit napsat jakýkoliv character, avšak zaznamenána budou pouze v případě, že se bude jednat o mezeru, všechna písmena a další speciální znaky. Do budoucna lze ještě více omezit.

Pokud však držíme shift, stisknuté klávesy nebudou zaznamenány z důvodu, že pokud držíme shift, jedná se o pohyb. A tudíž není optimální tyto stisky zaznamenávat. Převádění na malá písmena je pro budoucí použití komparace textů. Následný text vypisuje do přirazeného textového bloku.

Tato metoda je využívána v metodě Update().

### Update()

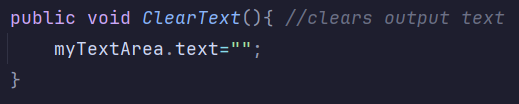
Update je volán každý snímek, takže pokud potřebujete číst vstup hráče, je to ideální funkce k tomu, abyste ho zpracovali, abyste nezmeškali žádnou událost. [12]

Obsah obrázku text, snímek obrazovky

Popis byl vytvořen automaticky

Kód 7: Metoda Update() tířdy WritingGameplay

Nejdříve od hráče získáme stisky na klávesnici, ty následně projedeme přes metodu BuffWord() (vysvětlena výše). Poté kontrolujeme z Control Words, jestli máme zastavit časování.  
V případě nezastavení časovače se nám textové pole po určitém čase (v tomto projektu po 3 s – nastaveno externě v editoru) přemaže. To zajišťuje metoda ClearText(). Zároveň se i časovač vynuluje a může znovu počítat.



Kód 8: Metoda ClearText() třídy WritingGameplay

Metoda ClearText() je nastavena na public, jelikož je využívána i jinými skripty.

## Následování textového pole postavy

Nad postavou bylo potřeba textové pole, do kterého se průběžně vypisuje to, co hráč zadá. Nešlo však vložit do objektu postavy – s otáčením se otáčelo taky. Tudíž bylo třeba jej dát mimo objekt postavy a naprogramovat chování.

Obsah obrázku text, snímek obrazovky, Písmo, Grafika

Popis byl vytvořen automaticky

Obrázek 2: Ukázka textového pole and characterem v Unity editor

Obsah obrázku text, snímek obrazovky, software, Písmo

Popis byl vytvořen automaticky

Kód 9: Třída FollowTarget

Tato třída se stará o následování objektu (v případě Tutorial levelu hráče). Je do ní vložen prvek characteru – target. Externě lze i upravit odchylky. SmoothDamp() vytvoří plynulý přechod z jedné polohy do druhé.

## Chování kamery

Kamera v Tutorial levelu následuje postavu a postupně se oddaluje a přibližuje na základě vzdálenosti kontrolních bodů.

### MoveCam()

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 10: Metoda MoveCam() třídy CameraMoveFollow

Stejně jako při textovém poli následujeme target (opět character). Avšak kamera následuje pouze horizontální polohu. Tudíž v případě, že postava skáče, kamera se na vertikální rovině nepohne. V případě, že character neskáče, kamera následuje i jeho vertikální polohu. Opět dokážeme externě v editoru upravit hodnoty offsetů, avšak tentokrát se nám o ně stará i metoda Remap().

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 11: Metoda Remap() třídy CameraMoveFollow

Tato metoda přemapuje vstup na nový scale. Parametry jsou aktuální hodnota, kterou chceme přemapovat, po té hodnoty, ze kterých mapujeme, a po nich nové rozmezí. Původně jsem se snažila funkci sepsat sama, byla však velice neefektivní, tak jsem využila kód z Unity fóra. [13]

### ZoomCam()

Tato metoda se stará o přibližování/oddalování kamery na základě dvou daných bodů. Tyto body jsou umístěny ve scéně a do skriptu jsou předávány jako reference. Body lze volně pohybovat, lze určit maximální a minimální hodnotu přiblížení.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 12: Metoda ZoomCam() třídy CameraMoveFollow

## Interakce s objekty scény

Interakce jsou vedeny díky kolizí. Jamile postava koliduje s objektem, spustí se akce.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 13: Příklad kolize při vstupu

V tomto případě využíváme kolize při vniku. Kolize se tak zaregistruje pouze jednou (vícekrát v případě, že bychom objekt opustili, a následně do něj opět vešly).

### Vypisování textu

Po úspěšné kolizi se spustí vypisování textu do prostředí. Text je zadán manuálně v editoru, vypisuje se znak po znaku do daného textového prostředí.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 14: metoda Update() třídy WriteText

### Interakce

Při kolizi se přepne, které textové pole hráč využívá. Základní možnost je textové pole nad postavou. V případě kolize se však přepneme na využívání textového pole kolidovaného objektu. V případě opuštění kolize se opět přepínáme na základní textové pole.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 15: Logika kolizí třídy InteractText

V kolizích jsou využívány metody ChangeTextArea() a UseDefaultTextArea() třídy WritingGameplay.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 16: Metoda ChangeTextArea() třídy WritingGameplay

Do funkce je poslána informace, jaké textové pole chceme využívat. Tím se ve WritingGameplay změní odkaz na aktuálně používané textové pole.

Obsah obrázku text, Písmo, snímek obrazovky

Popis byl vytvořen automaticky

Kód 17: Metoda UseDefaultTextArea() třídy WritingGameplay

Tato metoda zajistí přepsání ukazatele na původní textové pole (textové pole charaktera).

Ve třídě využívám jednoduchý Enum pro snadnější orientaci při vracení výsledků. Tyto hodnoty jsou využívány v metodě CheckIndexes().



Kód 18: Jednoduchý Enum třídy InteractText

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 19: Metoda CheckIndexes() třídy InteractText

Shodu textu pro interakci a zadaného textu uživatelem kontroluje metoda CheckIndexes(), která kontroluje každý zadaný znak. Následně vrací, zda se texty shodují, nebo se jedná o prázdný řetězec.

Obsah obrázku text, Písmo, snímek obrazovky

Popis byl vytvořen automaticky

Kód 20: Příklad využití hodnoty v porovnání s hodnotou Enumu

Pokud se oba texty shodují, daný objekt vyšle signál pro změnu textového pole na původní a poté se smaže.   
V případě neshody se zadaný text vymaže a uživatel musí začít znovu.

Obsah obrázku text, snímek obrazovky, Písmo, Grafika

Popis byl vytvořen automaticky

Obrázek 3: Ukázka interakce – text k interakci zašedlý, text hráče bílý

# Entre level

Entre level je čistě o vypisování textu na obrazovku. Tyto texty jsou uloženy v textovém souboru, ze kterého jsou při spuštění levelu načteny a postupně vypisovány.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 20: Metoda LoadStrings() třídy WriteLines

Textový soubor je vložen skrze třídu TextAsset. Tato třída je definovaná Unity. V Unity totiž nelze použít standartní způsob používání souborů. Problém totiž nastává při buildu projektu. V tuto chvíli zanikají všechny cesty, tudíž k souboru se nedá již dostat, tudíž nutno použít třídy Unity. Obsah daného TextAssetu se vypisuje jednoduše metodou ToString().  
Získaný text oddělíme na samostatné řádky. Tyto samostatné věty pak uložíme do Listu (c# třída se stejnými vlastnostmi a chováním jako ArrayList) \_sentenceList.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 21: Metoda PrintSentences() třídy WriteLines

Metoda PrintSentences() se stará a vypisování vět písmeno po písmenu z daného Listu seznamu vět. Tato metoda je speciální tím, že se nechová jako standartní metoda. Je třídy IEnumerator, která je využíváná Coroutines.

### Coroutines

Coroutine vám umožňuje rozložit vykonávání přes několik snímků, aby se jednotlivé snímky nezatěžovaly velkými výpočty. [14]  
Jedná se o fuknci, která může pozastavit své vykonávání (yield) až do doby, kdy je dokončena daná YieldInstruction. [15]

### IEnumerator PrintSentences()

V této metodě vypisujeme znak po znaku. Po každém vytištěném písmenu se čeká po dobu ručených sekund. To stejné nastává po dokončení celé věty.  
yield return new WaitForSeconds() zařizuje to, že program se zastaví, dokud neuplyne daný počet sekund (tyto hodnoty udány externě v Unity editoru). Po uplynutí daných sekund program opět pokračuje z místa, kde se zastavil.

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky

Obrázek 4: Příklad vlastností objektu ve scéně

# Kódová stránka prologu

Prolog slouží k většímu prohloubení mechanik naučených v Tutorial levelu. Využíváme stejný pohyb kamery a její postupné zoomování. Tentokrát však nevyužíváme možnost, že od interagovaného objektu lze odejít. Hráč musí prvně dokončit interakci, až poté může pokračovat.  
Text je do textových polí vypisovat stejně jako v Entre ze souborů. Je však řízen i interakcí. Ty fungují stejně jako v Tutorial levelu, kontrolují znak po znaku. V případě zadání správné věty se zpustí výpis další věty.

Obsah obrázku text, Písmo, snímek obrazovky, bílé

Popis byl vytvořen automaticky

Obrázek 5: Hierarchie interagovaného objektu

Objekt má přiřazen vlastní Canvas ve kterém se nacházejí dvě textová pole – jedno pro výpis ze souboru (Stable) a druhé pro input hráče (Input).  
Objekt writeText vypisuje věty ze souboru do textového pole Stable.  
Objekt interactText kontroluje, zda se zadaný řetězec shoduje s řetězcem pro interakci.  
Samotné psaní je řešeno na MainCharacter. Znaky zadány zde se vypisují v textovém poli Input. Objekt postavy využívá stejný script jako v případě Tutorial level.

## Přecházení na další levely

Přecházení na další levely je řešeno přes kolize. Jakmile kolidujeme s objektem, který má přepínání na starost, zjistíme index aktuálního levelu a spustíme následující level buildu.

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

Popis byl vytvořen automaticky

Kód 22: Systém přepínání na další level

Obsah obrázku text, skica, design, umění

Popis byl vytvořen automaticky

Obrázek 6: Ukázka Prologu – výpis textu ze souboru

# Levely uvnitř domu

Levely Kitchen, LivingRoom, Bedroom fungují na stejném způsobu jako levely doposud. Pouze kamera je tentokrát statická.

Obsah obrázku text, snímek obrazovky

Popis byl vytvořen automaticky

Kód 23: Třída pro přechod levelů

Do třídy v editoru vkládáme objekt, který chceme aktivovat. ControlWordsProlog se odkazujeme do slov, které řídí level.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 24: Třída ChangeSettingsInControlWords

Tuto třídu napojíme na prvek, díky kterému chceme měnit nastavení. V našem případě se jedná o objekt u postele. Při kolizi s ním zaktivujeme objekt u PC a kolizí s ním tak můžeme přejít na další level. (Objekt pro přechod na další level je defaultně vypnut).

# Boss Level

Tento level se zásadně liší od levelů doposud. Tentokrát již nehýbeme s hráčem, zůstaly nám pouze mechaniky psaní. Pro výpis textů ze souboru využíváme opět stejný princip.

Motivem tohoto souboje je konfrontace Milanova otce ohledně nedávných událostí – smrt Milanovy matky. Milan si je jistý, že otec ji zabil, avšak otec se snaží hodit vinu na něj.

Tento level má dva možné výsledky. Záleží čistě na hráči, jak si v souboji povede.

## Systém boje

Část textů hráč opět pouze opisuje. Tyto texty nemají vliv na výsledek. Avšak v určitých částech si hráč musí vybrat, které ze třech slov použije. Od těchto voleb se poté odvíjí výsledek souboje.

V této části máme dva typy souborů:

* Standartní texty na výpis (obsahuje pouze jednu větu)
* Texty pro boj (obsahuje tři slova)

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky Obsah obrázku text, snímek obrazovky, Písmo, software

Popis byl vytvořen automaticky

Obrázek 6: Příklad standartního textu Obrázek 7: Příklad textu pro boj

Výpis standartních textů následuje vzor výpisů doposud: text je vypsán na obrazovku, uživatel jej napíše, pokračuje se dále.

## Práce s bojovými slovy

Pro používání bojových slov a práci s nimi je vytvořena vlastní třída ManagingWords.  
Tato třída se stará o vypsání slov do daných textových polí. Obě metody jsou používány externě, tudíž nutný signature public.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 24: Metody pro nastavení a vymazání textu

Tato třída také obsahuje metodu pro zjištění délky nejdelšího prvku. To je později využíváno při interagování – při překročení počtu znaků nejdelšího se zadaný text automaticky vymaže.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 25: Metoda pro vrácení velikosti nejdelšího prvku

## Řízení pořadí vypisování

Každý objekt má dané své číslo pořadí. Třída s kontrolními slovy obsahuje hodnotu aktuálního indexu pro výpis. Jestliže se tento index rovná indexu objektu, daný objekt může plnit svou funkci.

Tato funkce je využívána v Coroutines, kde se čeká, dokud daný objekt nebude na řadě.

## Váha slov

Každé slovo, které je v bojovné části má předem stanovenou váhu – kolik HP ubírá bossovi. Tato slova jsou odděleně zapsaná v souboru společně s jejich váhou.

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Popis byl vytvořen automaticky Obsah obrázku text, Písmo, snímek obrazovky

Popis byl vytvořen automaticky

Obrázek 8: Příklad bojových slov a jejich vah Kód 26: Konstruktor pro object třídy Word

Z tohoto souboru jsou potom ve třídě WorkingWords čtena a je z nich tvořen List objektů třídy Word.

Obsah obrázku snímek obrazovky, text, Písmo

Popis byl vytvořen automaticky

Kód 27: Převádění slova na object Word a jeho následné vložení do Listu

## Třída WriteIntoBubble

Tato třída se stará o procházení bojových slov a porovnávání jich se slovem zadaným.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Kód 28: Kontrola jesli byl zadán text pro interakci

### V této části kódu jsou nejpodstatnější:

controls.dadHp -= listWord.Damage;

V kontrolních slovech je určeno HP bosse. V případě správného zadání slova se HP odečte daná váha slova.

Move()

Jedná se o metodu, která průběžně posouvá Canvas. Velikost posunu je dána počtem vět předcházejícího objektu (dle počtu vět je volena velikost bubliny, do které budeme vypisovat – děláno ručně, šlo by zalgoritmizovat).

Pro posun využíváme ručně zjištěné velikosti posunu na základě velikosti bubliny.

Obsah obrázku text, Písmo, snímek obrazovky, typografie

Popis byl vytvořen automaticky

Kód 29: Vytvořené konstanty pro posun a jejich hodnoty

Ve funkci Move() tyto konstanty využijeme pro nastavení hodnoty posunu dle počtu vět předchozího objektu. Tuto hodnoty poté pošleme jako parametr metody MoveUp() třídy MoveScreenUp.

Využíváme zde stejné logiky jako při následování objektu.

Obsah obrázku text, Písmo, snímek obrazovky

Popis byl vytvořen automaticky

Kód 30: Metoda MoveUp() třídy MoveScreenUp

## Ukázky ze souboje

Obrázky jsou pořízeny v Unity editoru, proto lze vidět různé rušivé prvky (označení textových polí, zvuku). Ve finálním buildy však tyto ukazatele nejsou.

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky

Obrázek 9: Standartní psaní textu

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky

Obrázek 10: Volba textu

## Různé možnosti konce

Hráč vyhraje, pokud se mu podaří dostat HP otce na nulovou, či zápornou hodnotu. V tom případě se nám otevře vítězný level, ve kterém si Milan uvědomí, že se otce nemusí bát.

V případě, že HP otce zůstane kladné, prohráváme a přepneme se na level, kde Milan panikaří, že jej jde otec zabít.



Kód 31: Přepnutí levelu dle zbývajícího HP otce

# Další prvky hry

## Main menu

Jedná se o úvodní menu hry. Zobrazuje název hry, možnost hru zapnout, nahlédnout do ovládání, ukončit hru.

## Pause menu

Jako každá hra, i tato hra jde zastavit. V Pause Menu lze hru obnovit, jít zpět do hlavního menu, nahlédnout do ovládání, ukončit hru.

Při otevření menu hru zastavíme pomocí zastavení času: Time.timeScale = 0f;  
Po zavření menu čas opět obnovujeme: Time.timeScale = 1f;

# Potíže při vývoji

## Nefungující rigování

Již ze začátku vývoje jsme narazili na zvláštní problém. Při animování postavy nám nešlo hýbat s kostmi postavy. V té době jsme používaly nejnovější verzi Unity. Avšak po pročtení si pár fór jsem přišla na to, že daná verze má bug právě v nefungujícím rigováním. Tím, že jsme byly teprve na začátku vývoje, rozhodly jsme se změnit verzi editoru. Tím se problém vyřešil.

## Nelze zvolit položky Pause menu v levelech 5 a 6

Jedená se o levely LivingRoom a Bedroom. V těchto levelech nelze zvolit položky Pause menu. Nejspíše se jedná o problém v hierarchii vrstev. Avšak menu jsem nastavila na hodnotu 500, čehož by žádný jiný prvek nikdy dosáhnout neměl, ale i přes to to nefunguje.

## Špatný výpis řetězců pro interakci

Tento problém nastal při vývoji nečekaně. Script nějakou dobu fungoval, avšak při jednom použití se objekt daného skriptu rozbil a tím rozbil i ostatní objekty. Ne však všechny. Některé objekty stejného Prefabu (objekt scény uložen do složky projektu pro budoucí použití) fungují správně a jiné ne. První větu vždy vypíší správně, ale následně vždy první slovo věty je první písmeno větu předchozí, avšak zbytek věty je již správně. Nejzvláštnější na tom však je, že další věta vždy vypíše to správné první písmeno věty předchozí, nikoli to, co se vytisklo.

|  |  |
| --- | --- |
| Správné texty | Vypsané texty |
| I said | I said |
| She is | Ihe is |
| We went | Se went |
| They hid | Whey hid |

Tabulka 2: Přehled textů

Jedná se nejspíše o interní problém v pointerech. V aktuální chvíli danou situaci projednávám na online fórech.

Škoda je, že zrovna tento problém neskutečně ovlivňuje hratelnost výsledné hry.

# Nevyužité nápady

Nevyužitých nápadů bylo mnoho. Proto zde pouze zmíním ty, které měly potenciál.

## Systém hydratace

Původním nápadem bylo do hry implementovat režim hydratace. Hráč by hydrataci musel postupně během hry doplňovat, jinak by se objevily nepříjemné efekty: nahrazování zadaných písmenek písmeny náhodnými. Což ve hře, která je založena na psaní textu, je podstatný problém.

Nápad jsem však opustila z prostého důvodu: tato mechanika do hry nic nepřinesla, pouze by hráče frustrovala. Její implementace tudíž byla zavržena.

## Papoušek jako mazlíček

Milan měl mít původně papouška jako mazlíčka. Mohl by jej učit různé triky. Již jsme měly nápad na dvě minihry: učení papouška trikům a učit jej mluvit. Možnost s mluvou nám do daného tématu seděla více, avšak po zrušení spolupráce s původním grafikem mi již neseděla do tématu.

## Možnost používání mobilu

Původním plánem bylo do hry zakomponovat více moderního světa. A jak lépe toho dosáhnout než přes využití mobilu. Mobilem mělo jít fotit a hrát na něm různé hry. Od tématu jsem opustila ze stejných důvodů jako o téma výše.

## Původní systém psaní

Původní systém psaní se lehce lišil od toho aktuálního. A to v tom, že původně se slova pro interakce neměla vůbec zobrazovat. Hráč by daná slova získal sledováním okolí a příběhu, ta základní z příručky připnuté v menu.

Od tohoto nápadu jsem opustila po konzultaci s bratrem, kdy jsme se pohádali o tom, jak lze použít X slov pro jednu věc. A toto je věc, co se vám nechce programovat. Ošetřovat každé jedno slovo dané situace. Navíc tento systém byl hodně nefér ke hráči, tudíž jsem jej opustila.

# Budoucí vývoj

Maturitní prací tento projekt jistě nekončí. Již teď mám v hlavě nápady, které jsem buď chtěla a nestihla zrealizovat, nebo je plánuji do budoucna.

## Lepší interakce

Již v této verzi hry jsem chtěla mít lepší interakce. Například v kuchyni: lednice by se dala otevřít, ze dřezu bychom vytáhli hrnek. Grafické assety k tomu mám již nachystané, bohužel však nevyšel čas na jejich implementaci.

## Sbírání knih s příběhem

Původním plánem bylo do hry zahrnout příběh, který se nachází v sekci **Nový příběh**. Úryvky textu by hráč postupně nacházel v domě v podobě knih. Ty by poté mohl přečíst v menu a dozvědět se tak události, které předcházely dění hry.

## Možnost uložení a načtení hry

Při budoucím vývoji bude jistě na místě přidat možnost uložení aktuálního pokroku, popřípadě jeho načtení. Avšak s povahou hry bych implementovala spíše automatické uložení.

## Kosmetické úpravy

Hru bude potřeba celkově graficky doladit. Přidat přechody mezi levely, přidat vice animací.

Pokyny pro hráče

# Citovaná literatura

1. **Tišnovský, Pavel.** Historie vývoje počítačových her (1.část - první milníky) - Root.cz. *Root.cz.* [Online] 10 11 2011. [Cited: 20 12 2023.] https://www.root.cz/clanky/historie-vyvoje-pocitacovych-her-1-cast-prvni-milniky/.

2. **Gaming: Trendy, význam a postoj Čechů. *Erste Group - Home | Erste Group Bank AG.* [Online] [Cited: 2 11 2023.] https://www.erstegroup.com/en/research/report/cz/SR355728.**

**3. Herní žánry na Databázi her – Nápověda – Databáze-her.cz. *Databáze-her.cz – domov všech hráčů videoher.* [Online] [Cited: 20 12 2023.] https://www.databaze-her.cz/napoveda/herni-zanry-na-databazi-her/.**

**4. Co je to herní engine [Obecné téma] - CESKEMODY.cz . *CESKEMODY.cz - MODIFIKACE A MAPY PRO POČÍTAČOVÉ HRY.* [Online] [Cited: 20 12 2023.] https://www.ceskemody.cz/clanky.php?clanek=56.**

**5. [2023] The Most Used Game Engines: A Comprehensive Guide - Stack Interface. *Stack Interface.* [Online] [Cited: 20 12 2023.] https://stackinterface.com/most-used-game-engines/.**

**6. Unity at 10: For better—or worse—game development has never been easier. *Ars Technica.* [Online] [Cited: 2 1 2024.] https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/.**

**7. C Sharp – Wikipedie. *Wikipedie, otevřená encyklopedie.* [Online] [Cited: 20 12 2023.] https://cs.wikipedia.org/wiki/C\_Sharp.**

**8. Rider. *JetBrains: Essential tools for software developers and teams.* [Online] [Cited: 20 12 2023.] https://www.jetbrains.com/guide/tags/rider/.**

**9. GitHub – Wikipedie. *Wikipedie, otevřená encyklopedie.* [Online] [Cited: 20 12 2023.] https://cs.wikipedia.org/wiki/GitHub.**

**10. About GitHub Desktop - GitHub Docs. *GitHub Docs.* [Online] [Cited: 20 12 2023.] https://docs.github.com/en/desktop/overview/about-github-desktop#about-github-desktop.**

**11. Update vs. FixedUpdate vs. LateUpdate in Unity - LogRocket Blog. *LogRocket Blog.* [Online] 22 1 2024. https://blog.logrocket.com/update-vs-fixedupdate-vs-lateupdate-in-unity/#fixedupdate.**

**12. Update vs. FixedUpdate vs. LateUpdate in Unity - LogRocket Blog. *LogRocket Blog.* [Online] [Cited: 20 1 2024.] https://blog.logrocket.com/update-vs-fixedupdate-vs-lateupdate-in-unity/#update1.**

**13. Re-map a number from one range to another? *Unity forum.* [Online] 22 1 2024. https://forum.unity.com/threads/re-map-a-number-from-one-range-to-another.119437/.**