

Střední průmyslová škola strojní   
a elektrotechnická a Vyšší odborná škola, Liberec 1, Masarykova 3

Unity VR hra

Maturitní práce

Autor **Filip Margitai**

Obor **Informační technologie**

Vedoucí práce **Mgr. Michal Stehlík**

Školní rok **2021/2022**

Anotace

Práce se zabývá vývojem hry ve virtuální reality (VR) za použití 3D enginu Unity.

Přináší pár hodin zábavy, ale také spoustu řešení na různé problémy, týkající se vytváření VR her v Unity, pro všechny studenty, učitele ale i veřejnost prostřednictvím repozitáře.

Summary

This work …

Čestné prohlášení

Prohlašuji, že jsem předkládanou maturitní práci vypracoval sám a uvedl jsem veškerou použitou literaturu a bibliografické citace.

V Liberci dne

Filip Margitai

Obsah

[Úvod 1](#_Toc96970042)

[1 Teorie o VR 2](#_Toc96970043)

[1.1 Historie 2](#_Toc96970044)

[1.1.1 Sci-fi 2](#_Toc96970045)

[1.1.2 Vznik 2](#_Toc96970046)

[1.1.3 Rozšíření 3](#_Toc96970047)

[1.1.4 Přítomnost 3](#_Toc96970048)

[1.2 Jak funguje VR? 5](#_Toc96970049)

[1.1.5 Brýle 5](#_Toc96970050)

[1.1.6 Ovladače 5](#_Toc96970051)

[1.3 Využití VR 5](#_Toc96970052)

[1.1.7 Videohry 5](#_Toc96970053)

[1.1.8 Stavebnictví 5](#_Toc96970054)

[1.1.9 Řízení (armáda, autoškoly) 5](#_Toc96970055)

[2 VR v Unity 6](#_Toc96970056)

[3 Moje práce 7](#_Toc96970057)

[3.1 Stručný popis 7](#_Toc96970058)

[3.2 Základní ovládání 7](#_Toc96970059)

[3.3 Inspirace a nápad 7](#_Toc96970060)

[3.3.1 Papers, please 7](#_Toc96970061)

[3.3.2 Prostředí – Rick a Morty, Muži v černém 7](#_Toc96970062)

[4 Modely v Blenderu 9](#_Toc96970063)

[4.1 Seznámení s prostředím 9](#_Toc96970064)

[4.2 První modely 9](#_Toc96970065)

[4.3 Modely do hry 9](#_Toc96970066)

[4.4 Textury 9](#_Toc96970067)

[5 Problémy a jejich řešení 10](#_Toc96970068)

[5.1 Základní ovládání 10](#_Toc96970069)

[5.1.1 Chůze a ruce 10](#_Toc96970070)

[5.1.2 Chytání předmětů 10](#_Toc96970071)

[5.2 Sandbox (pískoviště) 10](#_Toc96970072)

[5.3 Generování dokladů 10](#_Toc96970073)

[5.3.1 Původní kód 10](#_Toc96970074)

[5.3.2 Nakreslení problému 10](#_Toc96970075)

[5.3.3 Navržení podle diagramu 10](#_Toc96970076)

[5.3.4 Náhodné generování platných i neplatných dokladů 10](#_Toc96970077)

[5.3.5 Seznamy dat 10](#_Toc96970078)

[5.3.6 Razítka 10](#_Toc96970079)

[5.4 Chod a kontrolování hry 10](#_Toc96970080)

[5.5 Bugy a jejich řešení 10](#_Toc96970081)

[5.5.1 Bug 1 10](#_Toc96970082)

[5.5.2 Bug 2 10](#_Toc96970083)

[Závěr 11](#_Toc96970084)

[Seznam zkratek a odborných výrazů 12](#_Toc96970085)

[Seznam obrázků 13](#_Toc96970086)

[Použité zdroje (1) 14](#_Toc96970087)

[A. Seznam přiložených souborů I](#_Toc96970088)

Úvod

Práci jsem si vybral na základě mých předchozích zkušeností, jak v enginu Unity, tak s virtuální realitou.

K práci je použit Oculus Quest 2 od společnosti Meta (Facebook).

# Teorie o VR

Technologie virtuální reality umožňuje změnit pohled uživatele a přenést jej do jakéhokoliv prostředí, jenž si dokážeme představit. Podle dané aplikace se člověk může ocitnout ve světě relaxačním, ve kterém chodí a kochá se, ve světě herním, ve kterém se aktivně hýbe a snaží se splnit daný cíl hry, či ve světě simulačním, ve kterém se může například učit pilotovat letadlo. Virtuální svět nemusí být vždy celý počítačově generovaný ale může generovat virtuální předměty do kamerou natáčeného reálného světa – toto odvětví nazýváme rozšíření reality (AR) a stalo se natolik populární, že jej nepovažujeme za poddruh virtuální reality, nýbrž za svoji vlastní kategorii, z tohoto důvodu jí mimo toto zmínění nebudeme v této práci rozebírat.

## Historie

Je důležité pochopit, že vývoj VR není záležitostí posledního roku, ale že je tu v primitivnějších a více experimentových formách už nějaké to desetiletí. Vraťme se tedy na úplný začátek a projeďme si pár klíčových momentů, které vytvarovaly dnešní svět VR tak jak ho známe.

### Sci-fi

Předtím než se virtuální realita stala fyzickou záležitostí, byly zařízení a myšlenky na podobném principu popisovány v několika zejména sci-fi dílech. Damien Broderick ve svém díle Judas Mandala (1982) poprvé používá pojem „virtuální realita“, ale myšlenkově se jedná o velmi rozdílné zařízení, přirovnatelné spíše k filmu The Matrix (1999). Další autor – Jaron Lanier uvádí, že rozšířil termín „umělá realita“, který se mnohem více podobá dnešní formě virtuální reality.

### Vznik

První výrobek, který se alespoň velmi vzdáleně podobá funkčností dnešní virtuální realitě, je „The View-Master“ od vynálezce Williama Grubera už v roce 1939. Jednalo se o přehrávač fotografií, ve kterém mělo každé oko rozdílný pohled – iluze 3D prostoru.

Za další velký skok bychom mohli považovat první headsetový (uchycení na hlavě) displej. Ten vznikl v roce 1960 a umožňoval přehrávání videí ve 3D. Už v dalším roce byl k této verzi headsetu přidán senzor pohybu. Je ale pochopitelné, že v tomto roce nebyl tento výrobek určen pro aplikace virutální reality, nýbrž pro vojenské účeli, umožňoval tedy přenášení obrazu z kamery k operátorovi, jenž se otáčel a dostával informace, co je kolem kamery.

V roce 1987 společnost VPL (visual programming lab) začala prodej prvního headsetu (EyePhone $9400) a rukavic ($9000) virtuální reality. Další společnosti se brzy přidali a prodávali své produkty pod názvem, jež VPL uvedl – Virtuální reality. Od té doby se už této technologii jinak neříkalo.

### Rozšíření

První velké rozšíření mezi mainstream zapříčinila Sega v roce 1994, kdy vydali svou první VR konzoli „The Sega VR-1“. V následujícím roce vydalo Nintendo první přenosnou konzoli „The Nintendo Virtual Boy“. Ta se však pro černobílí obraz a časté selhávání neprodávala a nenásledoval jí žádný nástupce.

V roce 2010 začíná vznikat první „Oculus“, na ten je za dva roky uvedený kickstarter a za další dva roky jej kupuje společnost Facebook (nyní Meta). V témž roce oznámí společnost HTC vývoj jejich prvního headsetu – „HTC Vive“. Ten je vydán v dalším roce po boku Valve – jež představuje své virtuální prostředí „SteamVR“ a oznamuje vývoj jejich vlastního headsetu. Ten však přijde do prodeje až v roce 2019 pod názvem „Valve Index“. V roce 2016 vydává Sony „Playstation VR“ – headset určený k hraní her na tehdejší konzoli Playstation 4.

### Přítomnost

V dnešní době jsou VR headsety stejně cenově dostupné, jako počítače, či konzole a zákazník má již široký výběr na základě ceny, kvality, podpory softwaru apod. Nejpoužívanějším headsetem je „Oculus Quest 2“, který mimochodem já používám, jež se do své pozice dostal svoji dostupnou cenou, za obětování kvality. Nenechte se však mýlit, za svou až poloviční cenu v porovnání s například „HTC Vive Pro“ nabízí minimální, pro většinu lidí nepoznatelný rozdíl v přesnosti a je ve všech recenzích, co jsem viděl jednoznačně doporučován. Velkým konkurentem na trhu cenově dostupných VR se zde ještě nabízí „Reverb G2“ od společnosti HP.

Pokud má člověk možnost připlatit si, nabízí se zde již zmíněný „HTC Vive Pro“, či Valve Index, oba přinášející si své výhody i nevýhody.

## Jak funguje VR?

### Brýle

### Ovladače

## Využití VR

### Videohry

### Stavebnictví

### Řízení (armáda, autoškoly)

# VR v Unity

# Moje práce

Postřehová hra ve V této VR hře se ocitnete v roli hraniční kontroly dokladů, Vaším úkolem je určit, zda daný – náhodně generovaný občan, nebo mimozemšťan mají platné doklady, či nikoliv. Cílem hry je získání co nejvyššího skóre, než se Vám naplní metr chyb. Pokud špatně určíte platnost občanových dokumentů, dostáváte jeden strike. Při dovršení tří striků pro Vás hra končí, ale každý nový den se z Vašeho účtu odmaže jedna chybka. Každý den Vám bude nápodobně určen počet odbavení, jež musíte stihnout – jejichž nestihnutí Vás bude stát jeden strike a dva v pokročilejších dnech. Hra graduje rychlostí a dřív nebo později narazíte.

## Základní ovládání

Ovládání hry jsem ponechal, jak by tomu tak mělo u VR být, velmi minimalistické a s omezeným počtem možných inputů – z mých předchozích zkušeností s prací ve VR totiž vím, že uživatel (v tomto případě hráč) je poměrně hloupé stvoření a není schopný rozeznávat jasně rozeznatelná tlačítka na ovladačích VR. Z tohoto důvodu je primárním ovládáním trigger – jenž sebere předmět a více s předmětem, lze dělat pouze pohybem, či puštěním tlačítka. Zmíněný pohyb – například označení dokladu razítkem, je pro člověka mnohem přirozenější než mačkání dalšího inputu a stává se pro novodobější VR projekty samozřejmostí. Dalšími inputy jsou joysticky, na každé ruce určen k jiné činnosti – levá pro pohyb a pravá pro otáčení. Co se týče otáčení – implementoval jsem jej z pocitu důležitosti, ale později se dobral k závěru, že jen přidává na složitosti ovládání a ve výsledky spíše kazí herní zážitek. Proč ale? Otáčení je ve VR uskutečněno za pomocí orientace headsetu (hlavy). Z mých předchozích zkušeností jsem počítal, že není možno plného pohybu, kvůli drátovému spojení headsetu a počítače, s Quest 2 však plně běžím přes beta funkci – přenos přes wifi, který na mém rychlém domácím připojení běží bezproblémově. Ponechal jsem však tuto funkci, protože ne všechny platformy VR můžou běžet bezdrátově a taktéž, protože ne všude je dostatečně dobré bezdrátové připojení pro plynulý přenos obrazu.

## Inspirace a nápad

Ve hrách pro VR je často největší zábavou chaos. Například takový „Job Simulator“, kde spěcháte ale zároveň se snažíte dodržet všechny potřebná pravidla.

### Papers, please

Nápad kontroly pasů a dalších nezbytných dokladů přichází ze hry „Papers, Please“ (2013), ve které se vžijete do role hraniční kontroly dokladů, ve fiktivní zemi s nádechem socialistického režimu. Lidé přecházejí hranice za různými účely – od týdenní návštěvy příbuzných, přes několik měsíců práce, až po úplnou emigraci. Zdejším občanům, vracející se do země, stačí předložit občanský průkaz, zatímco lidé z cizích zemí musí u sebe mít potvrzení různého druhu, odvíjející se od jejich důvodu návštěvy a země, ze které přichází.

### Prostředí – Rick a Morty, Muži v černém

Moje hra se liší zasazením – tím je moderní svět roku 2067, ve kterém je nutno hlídat přílety a odlety na planetu Zemi z jiných planet. Inspirací pro toto zasazení byla hraniční kontrola v kultovním seriálu „Rick a Morty“ – 1. díl 1. série, kdy se hlavní hrdinové snaží propašovat zakázané ovoce skrze vesmírnou hraniční kontrolu.

Druhou inspirací je scénka z prvního filmu „Muži v černém“, kde dochází k hraniční kontrole různorodých mimozemšťanů, právě tak, jak k tomu dochází na našich existujících hranicích.

# Modely v Blenderu

V létě před třetím ročníkem jsem se v rámci toho, že jsem věděl, že budu chtít dělat 3D hru, začal učit v nejrozšířenějším modelovacím programu. Tím je Blender, který nabízí neuvěřitelnou škálu tutoriálu pro nejrůznější věci a to všechno zdarma díky přístupu vlastníka společnosti Michaela Fiska, jenž se k tomu staví podobně jako vlastníci Linuxu.

V programu jsem se učil buď hraním s různými možnostmi, nebo díky tutoriálům, zejména tedy zmíním youtubera „Blender Guru“, který má nespočet užitečných tutoriálů a skvělé série pro naprosté začátečníky, kterými jsem se toho opravdu mnoho naučil.

## Seznámení s prostředím

Možnosti Blenderu jsou neuvěřitelně rozsáhle, ale jelikož nejsem grafik a nijak po tom ani netoužím, většina mé práce v něm probíhala na modelování a animování (animace pomocí kostry). Blender je na pohled složitý, ale když se do toho člověk opře, tak v zásadě jednoduchý program, pokud neděláte profesionální grafiku, důležité je však znát několik důležitých zkratek, které Vám ohromně ulehčí život – sám například ani nevím kde se tyto věci dají naklikat myší. Důležitou zkratkou je „TAB“, kterým přepínáte mezi objektovým módem a editovacím módem. Objektový mód slouží k manipulaci celého objektu, nebo například při zvětšování/zmenšování. Naopak v editovacím módu upravujete jednotlivé body, linky, či plochy – vytahováním, zvětšováním, zmenšováním, rozdvojení (z jedné plochy uděláme více ploch, každou pak můžeme měnit zvlášť). Užitečné je pak pamatovat přepínání mezi body, linkami a plochy – „1“ „2“ „3“ v tomto pořadí, či zkratky „R“ pro rotaci, „CTRL“ + „R“ pro přidání linek (rozdvojení), „E“ pro vytažení, „S“ pro změnu velikosti, „M“ pro změnu pozice, „X“ pro změnu po ose X a to samé s osami Y a Z.

## První modely

Mým prvním modelováním jsem následoval tutoriálovou sérii již zmíněného „Blender Guru“, který zde vytváří donut. Na to jsem navázal sérií youtubera „Imphenzia“, jež má naopak skvělé tutoriály pro low poly modelování (jednoduché ale dobře vypadající), třeba zrovna jeho série na low poly charakter, kde jej potom i riguje (přidává kostru) a podle kostry animuje, byla velmi užitečná.

## Modely do hry

Do hry jsem pokračoval s třemi low poly charaktery: člověk, medúza, blob (slymák). Poslední přítomný ve hře je Ugandan Knuckles, kterého jsem stáhl z internetu. Ten je původem postavou z herní série Sonic, ale tato jeho „znetvořená“ verze se stala populární v roce 2017, díky meme komunitě a díky streamerovi Forsen, jenž s tímto modelem chodil ve hře „VR Chat“ a opakoval hlášku „Do you know da wae?“. Jeho známosti jsem využil a dal jej do hry pod vlastní planetou Uganda, ze které pochází.

Animování, ačkoli nevypadá jakkoliv úžasně, je nevděčná práce – velmi těžké a čas beroucí je dělat, aby pro lidské oko vypadaly přirozeně. Navíc pro aspoň trochu vypadající efekt je nutné humanoidním (blob má animace tvarované, nikoli tedy podle kostry) postavám dodávat kostru, z které poté vytváříte pohyby.

## Textury

Původní plán byl nakreslit postavám textury skrze Bledner, ale výsledek procesu nevypadal příliš dobře v minimalistickém provedení a jelikož jsem nechtěl trávit více času učením se sofistikovanějšího způsobu texturování, textury jsem zpětně dodal v Unity, jednobarevné. Z těmi jsem si vyhrál aby nevypadaly jednotně a ve finální verzi plní svoji funkci dostatečně.

Na stěny a prostředí jsem využil textury z balíčku z Unity Asset Storu, které vypadají futuristicky a mají dostatečné rozlišení pro příjemný pohled.

# Problémy a jejich řešení

Průběh od samotného začátku aplikace až po ladění těch nejotravnějších bugů, se kterými jsem měl tu čest se setkat.

## Základní ovládání

Pro základní ovládání jsem zvolil v dnešní době jednoznačně nejlepší knihovnu OpenXR. Ta je podporována vývojáři Unity, ale stejně tak je i například knihovna Oculus, přímo dělaná pro můj VR headset. OpenXR má ale tu výhodu, že je normalizovaná pro všechny velké VR platformy. Pokud by tedy někdo jiný s rozdílným VR headsetem chtěl zkusit mou aplikaci, měla by aspoň ze stránky inputů běžet bezproblémově. Navíc OpenXR nabízí mnohem více funkcí než jiné knihovny a pro samostatné VR vývojáře je bezkonkurečně nejlepší možností.

### Ruce

První věcí, do které jsem se pustil byly ruce. Je možné zobrazovat ruce ve hře jako například ovladač, či ruku s ovladačem, ale já jsem se vydal za mě tou nejlépe vypadající možností – pouze ruce. Jejich model jsem stáhnul z oficiálních stránek Oculusu, kde jsou nejen zdarma ke stažení, ale nabízejí i kostru modelu. S kostrou mám totiž mnohem jednodušší práci s animacemi – ty jsem udělal dvoje: stisknutí triggeru (horní tlačítko, na kterém má uživatel ukazováček) a stisknutí gripu (spodní tlačítko, na kterém má uživatel prostředníček). Pokud tedy zmáčknete některé z těchto tlačítek, Vaše ruka ve virtuálním světě odráží tvar Vaší ruky ve světě reálném. I přes to, že trigger není na žádnou funkci ve hře využit, dodává to pocit propojení mezi pohyby uživatele a virtuálním prostředím.

### Pohyb a otáčení

Na pohyb a otáčení jsem použil už předem naprogramovaný preset z OpenXR, to mi také zajistí sjednocení všech různých druhů joysticků, či trackpadů, jenž mají jednotlivé VR konzole rozdílné. Zahlcený programováním jsem si tedy ušetřil trochu práce, ale naopak jsem čas věnoval štelování s desítkami různých parametrů. Pro otáčení jsem nakonec zvolil postupné otáčení a postupný pohyb. To může spoustě uživatelům, hlavně těm, jež se nepohybují ve VR, zapříčinit mírné závratě, za mě se však jedná o mnohem pohodlnější ovládání, jelikož člověk neztrácí pojem o tom, kde je a nabízí to i možnost preciznějších pohybů a zejména otáčení, kterého byste se snapovacím ovládáním nikdy nebyli schopni dosáhnout.

### Chytání předmětů

Chytání předmětů a jejich následné skládání či házení je základ všech úspěšných VR záležitostí a kolikrát i nejzábavnější věcí v celé VR aplikaci, ačkoli se na to často ani nezaměřuje. Na to jsem využil další funkci OpenXR, která umožňuje tyto základní funkci, ačkoli teda s několika nepříjemnostmi. První takovouto nepříjemností je orientace – ta se vždycky řídí podle orientace rukavice, to však znamená, že předmět po sebrání může naprosto změnit svoje otočení v mžiku sekundy. To není až takový problém v mé hře, jelikož jedinými věcmi, jež potřebujete brát jsou papíry (dokumenty) a razítka. Otevřelo to ale cestu nepříjemnému bugu, který řeším v kapitole Bug 1.

U házení předmětů jsem nenarazil na žádnou nedodělanost a u předávání předmětu z ruky do ruky je opět nepříjemná změna orientace a funguje to taktéž na principu primární ruky – pokud držíte předmět v této ruce, musíte jej pustit, aby ho druhá ruka mohla vzít, v kontrastu s obráceným případem, kde stačí aby obě ruce držely grip tlačítko. Když jsem toto ovládání implementoval rok zpátky ve firmě Proconom, dělal jsem jej z nuly, a tak jsem měl možnost více si šáhnout na tyto problémy – otáčení fungovalo na principu změny, avšak změna probíhala v závislosti na původní orientaci předmětu. Co se týče předávání mezi rukami, pokud jakákoliv ruka uchopila předmět z ruky druhé, vždycky byl předán bez ohledu na primární ruku (z pohledu kódu neexistovala primární ruka). Tyto věci jsou sice pěkné, avšak mi tehdy zabraly okolo 20 hodin, což je porce času, jež v tomto projektu můžu věnovat důležitějším věcem. Na rozdíl od dřívějšího projektu nemám zde nutnost zvedání a otáčení předmětů. Kdybych však tuto hru dělal ve větším měřítku, určitě bych toto naimplementoval pro lepší pocit ze hry pro uživatele (game feel).

Je také důležité dát předmětům odpovídající hitboxy, což v případě nějakých předmětů jako je židle, může být obtíží. Naštěstí se moje hra na podobné předměty nezaměřuje a nemusel jsem tedy až na pár výjimek ručně obkreslovat hitboxy, aby odpovídaly tvaru předmětu.

## Sandbox

Postup, který ze svých předchozích projektů a obzvlášť z tohohle projektu můžu vřele doporučit je vytvoření sandboxu, neboli testovacího prostředí. Začátek vývoje většiny her spočívá v základech ovládání, než je však zkoušet už v rozpracovaném prostředí budoucí hry, je lepší si vytvořit prostředí, vhodné k testování nových funkcí a o to výhodnější k hledání bugů. V Unity, podobně jako ve většině herních enginů, vytvoříme scénu, vhodnou k testování a funkční objekty nadále přesuneme do už důležitějších scén pomocí prefabů.

V sandboxu jsem si tedy zkusil základní ovládání a vyhrál se s blbostmi jako například: kuželky, skládání předmětů na sebe, či braní těžších předmětů s větší nastavenou váhou.

## Generování dokladů

Největší hromadou kódu v mé práci je bezkonkurenčně generování dokladů. Na rozdíl od většiny kódu, který je zaměřený na fungování virtuálního prostředí, zajišťuje tento algoritmus náhodné generování osoby, dokladů a následně i chyb v dokladech, který má uživatel uhádnout.

### Původní kód

Moji největší chybou z celé práce bylo vrhnutí se do psaní algoritmu, bez rozvržení a bez představy o jak dlouhý algoritmus se jedná a kolik chyb v něm může nastat. Kód jsem začal psát takzvaně „od ruky“ a když bylo potřeba, vytvořil jsem odpovídající třídy, enum, nebo seznam dat. Když jsem se ale dostával v psaní kódu dál a dál, obzvláště v části generování chyb, data se zdála naprosto rozházená a jejich přístup se mohl rapidně lišit mezi třídami. Tento poslední bot byl hlavní příčinou toho, že jsem si řekl „takhle to nepůjde“ a zahodil jsem většinu přítomného kódu.

### Nakreslení problému

Rozhodl jsem se tedy nakreslit a pospojovat všechny důležité třídy a přidat jim podle šipek postup náhodného generování. Středem pozornosti je GameController, jenž rozhoduje o všem dění. K tomu mu napomáhají aktuální data = data se v průběhu hry mění a je tedy dobré mít přístup k aktuálním údajům, protože v průběhu generování osoby a jejich dokladů se na tyto data často šahá, mezi některé příklady patří například: aktuální šance (neplatného dokladu), seznam platných prací (pokud má osoba Pracovní povolení), či seznam hledaných zlodějů. Druhou důležitou datovou třídou je data collection = kolekce všech možných dat, od všech možností jmen, měst, prací atd. Z těchto kolekcí se náhodně generuje většina informací a zbytek je doplněn z aktuálních informací.

### Navržení podle diagramu

Algoritmus se hned psal o mnoho lépe, přístup k datům již nebyl tak kostrbatý a dědičnost zachránila předem nedosažitelné možnosti hlavně u všech možných dokladů, které dědí z abstraktní třídy doklad.

Rozdělení datové a funkční části pomohlo v orientaci v kódu, a hlavně v hledání chyb, kterých tu zde bylo o to méně.

### Náhodné generování platných i neplatných dokladů

Před vygenerováním dokladů se vždy vygeneruje osoba. Osobě jsou přidány náhodné údaje, odpovídající planetě, z jaké osoba přichází. Doklady se generují na základě přebrání údajů z osoby, aby na všech dokumentech byly stejné a nedošlo tak k neúmyslné chybě. Pokud pochází osoba z jakékoliv jiné planety než z naší Země, je jí vygenerován druhý dokument – nějaký ze čtyřech možností, podle důvodu příjezdu.

Nyní nastává fáze generování chyb. K tomu samozřejmě nastane jen, pokud je osobě určeno, že má neplatný údaj. Náhodně se tedy vybere, jaký neplatný údaj to je, což zapříčiní, že v obou dokladech může nastat jen jediná chybka. Samotná chyba je těžší na hledání, než kdyby zde bylo chyb více, a podporuje tak postupně se zvyšující obtížnost hry. Chyby můžou být různé, mezi nejjednodušší na naprogramování bylo: prošlé datum platnosti, chybí druhý doklad, při kterých se většinou jedná o jeden řádek, poměrně jednoduchého kódu. Naopak jiné jako například: jména nejsou stejná, město nepatří k planetě, či je hledaný zloděj, patří mezi složitější, kde jsem musel přistupovat do kolekce dat a brát z ní jen určitá data, či upravovat údaj na jednom z dokladů, aby v porovnání mezi sebou byly jiné.

### Seznamy dat

Seznamy jsou poměrně jednoduše vzaté. Od krátkých jako je seznam planet, přes delší jako například: seznam jmen, až po složitěji zapouzdřené jako je seznam měst. U těch je nutné pamatovat například, zda město vydává, nebo samozřejmě z jaké je planety. Pokud například nastane chybné město – je zde možnost, že město nesedí k planetě, což jde kolikrát poznat podle názvu, ale je tu i možnost, při které město nevydává. To se může stát velmi těžké k najití například u Země, kde víte, že to město existuje z vlastního života, ale nemůžete si být jisti, zda vydává. Podobně tomu tak je u planety LaleloAa, kde jsou všechny názvy měst složení slabik la, le, lo a můžou se tak snadno plést. Povolené práce můžou také velmi jednoduše zamotat hlavu, každý den se mění, takže je nutné hlídat si, zda den uběhl, ale zároveň některé se objeví více dnů po sobě, jelikož se vybírají náhodně a jejich množina je relativně omezená.

### Razítka

### Kniha s údaji

Hráč samozřejmě musí mít přístup k datům, jež kontroluje, například aktuální hledaní vězni, dnešní datum (pro prošlé dokumenty), nebo platná města (města, jež mají povolení vydávat doklady). Každý den je nutno aktualizovat v této knize údaje, jež se se dnem mění, ty se berou z objektu s aktuálními informacemi.

## Chod a kontrolování hry

Hra je ovládána objektem se třídou game controller. Jeho starostí je převážně reakce na hráčova rozhodnutí a reakce na uběhnutý čas. Po uplynutí času nastává nový den, jež přináší změnu některých údajů, jmenovitě hledané vězně, dostupné práce pro pracovní listy, aktuální den pro platnost dokladů, šanci na neplatnost dokladů.

Celá hra začíná novým dnem. Uživatel stiskne červené tlačítko pro dalšího člověka a jsou mu před obličej vygenerovány náhodné doklady. Uživatel je musí zkontrolovat a označit občanský průkaz buď zeleně (platí), nebo červeně (neplatí). Poté doklady zahodí na zem a pokud je jakkoliv označil, tak dostane zpětnou vazbu o správnosti jeho rozhodnutí, pokud je neoznačil, znovu se objeví a spadnou na stůl. Když ve hře uběhne stanovený čas, zatím nastavený na tři minuty, končí daný den. Hráč se však snaží všechno toto provést co nejrychleji, aby mu game controller nepřidělil strike za nedostatečný počet správně označených pasů. Proces se opakuje, dokud uživatel nenaplní tři striky. Game controller celou tuhle dobu dává pozor na striky a na hráčovo skóre.

## Bugy a jejich řešení

Během cesty jsem narazil na velkou škálu bugů (chyb), které jsou nedílnou součástí vývoje ať už her, tak jakékoliv jiné aplikace. Proto je nutné vědět jak se nimi prodrat a některé, jež jsem řešil déle, tu zvýrazním.

### Bug 1

Papíry (dokumenty) vždy přijmou rotaci ruky, namísto své rotace. Místo vytváření celého nového systému grabování, který by sice toto opravil, ale jak jsem již zmiňoval mohl by zabrat až 20 hodin, jsem problém vyřešil otočením viditelných předmětu v mateřském předmětu v prefabu. To znamená že zatímco hráč chytá a otáčí hitboxem (colliderem) parent objectu, jeho child objekty jsou ve správné pozici aby se dal dokument číst.

### Bug 2

Z nějakého důvodu se vytvoří pouze jeden ze čtyř dokumentů, tím je občanka. První dokument – imigrační list se mi povedl přidat, bylo předáván parametru do neexistující objektu a místo erroru či warningu se prefab zkrátka nevytvořil.

Závěr

Tak jsem se dostal až na konec.

Seznam zkratek a odborných výrazů

HTML

HyperText Markup Language – značkovací jazyk používaný pro tvorbu webových stránek.

Seznam obrázků

[Obrázek 1 Úplně bez legrace, mě tohle kotě docela děsí. 3](#_Toc86059903)

[Obrázek 2 Modré borůvky 3](#_Toc86059904)

Použité zdroje (1)

1. **History Of Virtual Reality . *Virtual Reality Society.* [Online] https://www.vrs.org.uk/virtual-reality/history.html.**

**2. Stehlík, Michal. *Návod k maturitním pracím 2020.* Liberec : Albatros, 2020.**

**3. Unity User Manual 2021.2. *Unity3d.* [Online] https://docs.unity3d.com/2021.2/Documentation/Manual/index.html.**

1. Seznam přiložených souborů

Na přiloženém datovém nosiči se nacházejí následující soubory a složky:

* **MP2010-Novák-Jan-L4-Tepelné\_čerpadlo.docx** – editovatelná verze dokumentace maturitní práce
* **MP2010-Novák-Jan-L4-Tepelné\_čerpadlo.pdf** – tisknutelná verze dokumentace maturitní práce
* **Výkresy** – kompletní výkresová dokumentace
* **Aplikace** – zdrojové kódy