

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**

**Fakulta elektrotechniky a informatiky**

Evidenčné číslo: FEI-183415-92579

# **Systém pre snímanie a vyhodnocovanie písania prostredníctvom tabletu**

**Diplomová práca**

Študijný program: multimediálne informačné a komunikačné technológie

Študijný odbor: informatika

Školiace pracovisko: Ústav multimediálnych informačných a komunikačných technológií

Vedúci záverečnej práce: doc. Ing. Radoslav Vargic, PhD.

**Bratislava 2023**

**Bc. Filip Škorec**



## ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Bc. Filip Škorec**  
ID študenta: 92579  
Študijný program: multimediálne informačné a komunikačné technológie  
Študijný odbor: informatika  
Vedúci práce: doc. Ing. Radoslav Vargic, PhD.  
Vedúci pracoviska: doc. Ing. Radoslav Vargic, PhD.  
Miesto vypracovania: Ústav multimediálnych informačných a komunikačných technológií

Názov práce: **Systém pre snímanie a vyhodnocovanie písania prostredníctvom tabletu**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

- 1) Získajte prehľad o možnostiach programovania prostredia virtuálnej reality so spojením so zvoleným grafickým tabletom.
- 2) Analyzujte spôsoby prepojenia VR zariadenia s grafickým tabletom.
- 3) Oboznámte sa so zberom dát z grafického tabletu v prostredí Unity.
- 4) Nakonfigurujte virtuálnu pracovňu v prostredí Unity pre využitie VR zariadenia a grafického tabletu.
- 5) Riešenie vyhodnoťte a zdokumentujte.

Rozsah práce: minimálne 2 AH a maximálne 3 AH okrem prípadnej ďalšej technickej dokumentácie

Termín odovzdania diplomovej práce: 12. 05. 2023  
Dátum schválenia zadania diplomovej práce: 08. 05. 2023  
Zadanie diplomovej práce schválil: prof. Ing. Gregor Rozinaj, PhD. – garant študijného programu

# Podakovanie

V prvom rade by som sa chcel poďakovať vedúcemu diplomovej práce doc. Ing. Radoslav Vargic, PhD za jeho odborné vedenie, pripomienky a trpezlivosť. Taktiež by som sa chcel poďakovať svojej snúbenici Márií Rajníkovej za pomoc a cenné rady. V poslednom rade by som sa chcel poďakovať rodine a kolegom za morálnu podporu a motiváciu.

# Anotácia

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Multimediálne informačné a komunikačné technológie
Autor:	Bc. Filip Škorec
Diplomová práca:	Systém pre snímanie a vyhodnocovanie písania prostredníctvom tabletu
Vedúci záverečnej práce:	doc. Ing. Radoslav Vargic, PhD
Miesto a rok predloženia práce:	Bratislava 2023

V súčasnej dobe je virtuálna realita čoraz viac rozšírenejšia v okruhoch verejnosti. Vďaka dostupnosti a cenových relácií sa čoraz viac využíva virtuálna realita v bežnom a akademickom živote. Hlavným prínosom je množstvo rôznych vizualizácií ale naďalej ostáva problém pokročilej interakcie so zobrazenými modelmi. Najčastejšie sa využíva priama interakcia pomocou dotyku rúk, prípadne špecifické gestá. Ďalšie možnosti poskytuje špecifický hardvér s podporou virtuálnej reality ako je špeciálna klávesnica a perá. Vďaka nášmu prepojeniu je možné simulovať písanie na plátno, papier alebo tabuľu, či už vo virtuálnej pracovni alebo učebni. V našom projekte sme sa zamerali na písanie vo virtuálnej realite na dvojrozmernom plátne, čo zahŕňa písanie textu, kresbu diagramov, modelov a iných pomôcok pri komunikácii. Tieto funkcie vieme umožniť pomocou prepojenia grafického tabletu s virtuálnou realitou. Vytvorili sme 3D virtuálnu miestnosť v prostredí Unity, ktorá slúži ako virtuálna pracovňa. V miestnosti sme ako vstupné zariadenie zbierali dáta z grafického tabletu a pera na samotné písanie, ale aj z hlavy a rúk používateľa aby sme umožnili interakciu s prostredím. Skúšali sme viacero metód písania čiar, z ktorých nám najviac vyhovoval komponent Line Renderer. Následne sme pridávali funkcionality ako písanie pomocou prsta, zmena farby písma, mazanie a ukladanie plátna. Taktiež sme vytvorili vlastné gestá na posun plátna a zmenu veľkosti.

Kľúčové slová: virtuálna realita, grafický tablet, interakcia človeka s počítačom

# Abstract

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Applied Informatics
Author:	Bc. Filip Škorec
Seminar paper:	Tablet writing and evaluation system
Supervisor:	doc. Ing. Radoslav Vargic, PhD
Place and year of submission:	Bratislava 2023

Nowadays, virtual reality is becoming more and more widespread in public circles. Thanks to its availability and affordability, virtual reality is increasingly being used in everyday and academic life. The main benefit is the number of different visualizations, but the problem of advanced interaction with the displayed models still remains. Direct interaction using the touch of hands or specific gestures is most often used. Additional options are provided by specific hardware with virtual reality support, such as a special keyboard and pens. Thanks to our solutions, it is possible to simulate writing on a canvas, paper or blackboard, whether in a virtual study or a classroom. In our project, we focused on writing in virtual reality on a two-dimensional canvas, which includes writing text, drawing diagrams, models and other communication aids. We can enable these functions by connecting a graphic tablet with virtual reality. We created a 3D virtual room in the Unity environment, which serves as a virtual workspace. In the room, we collected data from input devices such as graphics tablet and a pen for writing, but also from the user's head and hands to enable interaction with the environment. We tried several methods of drawing lines, of which the Line Renderer component suited us best. Subsequently, we added functionalities such as writing with a finger, changing the color of the font, erasing and saving the canvas. We also created our own gestures for moving the canvas and changing the size.

Keywords: virtual reality, graphic tablet, human computer interaction

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Virtuálna realita</b>	<b>2</b>
1.1 Využitie virtuálnej reality . . . . .	2
1.1.1 Vizualizácia VR . . . . .	2
1.1.2 Simulácie . . . . .	2
1.1.3 Hry . . . . .	3
1.1.4 Virtuálna pracovňa . . . . .	3
1.1.5 Virtuálna učebňa . . . . .	3
1.2 Vybrané možnosti interakcie vo VR . . . . .	4
1.2.1 Gestá . . . . .	4
1.2.2 Klávesnica . . . . .	5
1.2.3 Pero . . . . .	5
1.2.4 Tablet . . . . .	5
1.2.5 Grafický tablet . . . . .	6
1.3 Možnosti prostredia na vývoj VR aplikácie . . . . .	6
1.3.1 Unreal Engine . . . . .	6
1.3.2 CryEngine . . . . .	6
1.3.3 Unity Engine . . . . .	7
<b>2 Návrh a príprava riešenia</b>	<b>9</b>
2.1 Návrh riešenia . . . . .	9
2.2 Možnosti prepojenia VR zariadenia s grafickým tabletom a počítačom . . .	11
2.2.1 Prepojenie tablet s VR zariadením pomocou kábla . . . . .	11
2.2.2 Prepojenie tablet s VR zariadením pomocou Bluetooth . . . . .	12
2.2.3 Prepojenie tablet s VR zariadením pomocou počítača . . . . .	13
2.3 Inštalácie zariadení . . . . .	14
2.3.1 Inštalácia grafického tabletu na počítač . . . . .	14
2.3.2 Inštalácia Oculus Quest 2 na počítač . . . . .	15
2.4 Inštalácia prostredia a potrebných doplnkov . . . . .	16
2.4.1 Inštalácia Unity hub . . . . .	16
2.4.2 Inštalácia Unity editora . . . . .	16
2.4.3 Inštalácia Visual Studio . . . . .	16
2.5 Spojenie zmiešanej reality s virtuálnou . . . . .	17

2.5.1	Vuforia Engine . . . . .	17
2.5.2	Triangulácia . . . . .	18
2.6	Tvorba projektu . . . . .	19
2.6.1	Vytvorenie nového projektu v Unity Hub . . . . .	19
2.6.2	Inštalácia pluginov . . . . .	20
2.6.3	Prvé kroky v projekte . . . . .	20
2.6.4	Importovanie hlavy hráča do projektu . . . . .	20
2.6.5	Importovanie rúk hráča do projektu . . . . .	21
<b>3</b>	<b>Implementácia</b>	<b>22</b>
3.1	Zber dát . . . . .	22
3.1.1	Zber dát grafický tablet . . . . .	22
3.1.2	Zber dát hlava hráča . . . . .	23
3.1.3	Zber dát ruky hráča . . . . .	23
3.2	Spôsoby písania čiar v Unity . . . . .	23
3.2.1	Trail Renderer . . . . .	24
3.2.2	Paint Brush Tool . . . . .	24
3.2.3	UI Canvas . . . . .	24
3.2.4	Line Renderer . . . . .	25
3.3	Úpravy písma . . . . .	28
3.3.1	Úpravy písania - farba . . . . .	28
3.3.2	Úpravy písania - zmena nástroja . . . . .	29
3.3.3	Úpravy písania - premazanie plátna . . . . .	29
3.3.4	Úpravy písania - uloženie plátna . . . . .	29
3.3.5	Úprava plátna - škálovanie . . . . .	30
3.4	Gestá vo virtuálnom prostredí . . . . .	30
3.4.1	Vytvorenie a detegovanie gesta . . . . .	30
3.4.2	Úprava plátna pomocou gesta . . . . .	31
3.4.3	Pohyb plátna . . . . .	31
3.4.4	Škálovanie plátna . . . . .	32
<b>4</b>	<b>Diskusia</b>	<b>34</b>
<b>5</b>	<b>Záver</b>	<b>36</b>
	<b>Zoznam použitej literatúry</b>	<b>38</b>

<b>Prílohy</b>	<b>I</b>
<b>A Dokumentácia</b>	<b>II</b>
<b>B Zdrojový kód</b>	<b>IV</b>



# Úvod

V súčasnej dobe je virtuálna realita čoraz viac rozšírenejšia v okruhoch verejnosti. Vďaka dostupnosti a cenovým reláciám sa čoraz viac využíva virtuálna realita v bežnom a akademickom živote. Najčastejším prípadom týchto situácií sú virtuálne učebne, ktoré sa zameriavajú na spojenie viacerých osôb pripojených na veľké vzdialenosti.

Ďalším prípadom sú virtuálne pracovne zamerané na zefektívnenie práce jednotlivca. Umožňujú virtuálne pridávanie dodatočného hardvéru ako sú napríklad viaceré monitory ľubovoľných veľkostí alebo samotné premietanie modelov do 3D virtuálneho sveta. Používateľ virtuálnej pracovne je schopný pracovať v jednotnom prostredí bez ohľadu na to kde sa fyzicky nachádza.

Hlavným prínosom virtuálnej pracovne je množstvo rôznych vizualizácií, ale naďalej ostáva problém pokročilej interakcie so zobrazenými modelmi. Najčastejšie sa využíva priama interakcia pomocou dotyku rúk, prípadne špecifické gestá. Ďalšie možnosti poskytuje špecifický hardvér s podporou virtuálnej reality ako je špeciálna klávesnica a perá.

V našej práci sme sa zamerali na písanie vo virtuálnej realite na dvojrozmernom plátne, čo zahŕňa písanie textu, kresbu diagramov, modelov a iných pomôcok pri komunikácii. Tieto funkcie vieme umožniť pomocou prepojenia grafického tabletu s virtuálnou realitou. Vďaka nášmu prepojeniu bude možné simulovať písanie na plátno, papier alebo tabuľu, či už vo virtuálnej pracovni alebo učebni.

Získali sme prehľad o možnostiach programovania prostredia virtuálnej reality s grafickým tabletom a opísali sme ich v kapitole 1. Ďalej sme analyzovali spôsoby prepojenia VR zariadenia s grafickým tabletom a otestovali sme ich v podkapitole 2.2. Oboznámili sme sa s možnosťami zberu dát z grafického tabletu v Prostredí Unity a aplikovali ich v kapitole 3, kde sme tiež vytvorili virtuálnu pracovňu. Následne sme naše riešenie vyhodnotili a otestovali s dobrovoľníkmi v kapitole 4.

# 1 Virtuálna realita

Virtuálna realita, skrátene VR je interaktívne výpočtové prostredie, v ktorom sa vytvára predstava reálneho sveta. V tomto svete môže používateľ integrovať virtuálne (nereálne) predmety, javmi a prostredím.

Na dosiahnutie virtuálnej reality je nutné použiť špeciálne zariadenia ako sú VR headsety alebo iné zariadenia so senzormi pohybu človeka v priestore, ktoré umožňujú používateľovi pohybovať sa a interagovať s virtuálnym prostredím.

Primárnym cieľom virtuálnej reality je tvorba ilúzie, kde má používateľ pocit, že sa nachádza na nejakom mieste na svete. Tu používateľ interaguje s virtuálnymi objektami a prostredím pomocou vysokokvalitného audiovizuálneho premietania.

Virtuálna realita má veľa možností aplikácie od zábavného priemyslu a vytvárania hier až po vzdelávanie, medicínu, architektúru, umenie a podobné. Samotné využitie virtuálnej reality spolu so zariadeniami na jej sledovanie sa veľkou rýchlosťou rozvíja a je čoraz prístupnejšia a cenovo dostupnejšia širšej verejnosti.

## 1.1 Využitie virtuálnej reality

Virtuálna realita ponúka svoje využitie a výhody vo veľa priemysloch. Často sa virtuálna realita využíva na vizualizácie veľkého množstva dát, na simulovanie reálneho sveta na tréning profesii, na herný a zábavný priemysel, na vytvorenie virtuálnej pracovne v pohodlí domova, alebo na vytváranie učební pre študentov z celého sveta.

### 1.1.1 Vizualizácia VR

Jedným z hlavných prísunov virtuálnej reality je možnosť vizualizovať veľké a komplexné dáta pomocou modelov v 3D. Výhody sú najmä neobmedzený priestor na zobrazenie a pohľady z viacerých uhlov. Na bežné zobrazenie veľkého množstva dát sa využíva viacero monitorov. Pomocou použitia VR zariadenia nie je nutné využitie viacero fyzických monitorov, stačí vytvorenie virtuálnych a tým nadobudne zníženie nákladov[1].

### 1.1.2 Simulácie

Využitie virtuálnej reality sa rozvinulo aj do odvetvia simulácií. Vďaka simuláciám sa môžu používatelia vzdelávať a poznávať bezkontaktne a bez rizika reálneho poškodenia, či zranenia. Simulácie sú často delené do troch kategórií podľa úrovne realistickosti prostredia na nepohlcujúce, plne pohlcujúce a čiastočne pohlcujúce.

Nepohlcujúca virtuálna realita pracuje s prostredím tak, aby s používateľom neinteragovala až na samotnú postavu používateľa, či vytýčené aktivity. Do tejto kategórie často

zapadajú hry, kde je síce používateľ vo VR svete, ale nie je jeho stredobodom. Nepohlcujúca realita sa dá zabezpečiť použitím len VR headsetu, bez potreby dodatočného hardvéru.

Plne pohlcujúca virtuálna realita zabezpečuje realistický zážitok vo virtuálnom svete. Používateľovi dáva pocit prítomnosti vo VR svete ako keby sa nachádzal v skutočnom. Plne pohlcujúca virtuálna realita sa využíva na tréning profesii ako sú doktori, sestry, armáda, či letectvo bez ohrozenia zdravia používateľa a okolia. Na dosiahnutie takto vysokej úrovne virtuálnej reality je potrebný výkonný hardvér a taktiež zariadenia ako je VR headset, haptické rukavice, či telové kombinézy s detektormi zmyslov.

Čiastočne pohlcujúca realita je spojením nepohlcujúcej a plne pohlcujúcej reality. Vytvára formu 3D priestoru alebo virtuálneho prostredia, kde sa používateľ môže svojvoľne pohybovať a všetky interakcie sú smerované na používateľa. Interakcie sú len vizuálne bez reálneho fyzického vnímania. Táto forma virtuálneho prostredia sa najčastejšie využíva pri virtuálnych prehliadkach, či virtuálnych pracovných a učebných. [2, 3]

### **1.1.3 Hry**

Virtuálna realita sa so svojím pokročilým vývojom dostáva do povedania herného a zábavného priemyslu. Medzi nimi sú videohry považované za jeden z najväčších trhov pre technológiu VR, pretože tieto technológie je možné použiť na zvýšenie úrovne vnímania reality používateľmi. Dané hry môžu byť hrané pomocou mobilov, počítačov alebo herných konzol. S aktuálnou úrovňou hardvéru ponúkajú VR zariadenia herné zážitky s minimálnym oneskorením, či nevoľnosťou. [4]

### **1.1.4 Virtuálna pracovňa**

Virtuálna pracovňa je novým trendom pri práci z domu, ktorý je stále často spomínaný spolu s pandémiou Covid-19. Pri práci z domu má veľa ľudí problémy oddeliť pracovný život od bežného, či sa jedná o rovnaké prostredie alebo rozptyľovanie ostatnými ľuďmi. Práve tieto problémy rieši virtuálna pracovňa, ktorá vytvára separátnu pracovňu v domácom prostredí. Ponúka rozširiteľné pracovisko bez potreby dodatočného hardvéru ako sú monitory alebo periférie. Vďaka tomu vie používateľ zvýšiť svoju aktivitu a odbornosť[5].

### **1.1.5 Virtuálna učebňa**

Virtuálne vzdelávanie poskytuje novodobé prostredie v oblasti vzdelávania. Stáva sa novodobým trendom na súkromných a verejných školách v rôznych krajinách sveta. Vďaka možnosti aj online virtuálnej učebni je možné využitie vzdialeného spojenia pri nepriaznivých podmienkach ako je pandémia, či prírodné úkazy. Vyučovanie vo virtuálnej realite poskytuje prijateľné strojové rozhranie, ktoré je následne pochopiteľnejšie pre študen-

tov. Zameriava sa na interaktívne učenie a zlučuje praktické skúsenosti s vedomosťami študenta[6, 7].

## 1.2 Vybrané možnosti interakcie vo VR

Existuje mnoho druhov vstupov, ktoré vie používateľ využiť na interakciu s objektami vo virtuálnom prostredí, od hlasových povelov, cez gestá tela, až po vsupy z periférií. V najbližších sekciách sme analyzovali vybrané vstupy ako sú gestá, klávesnica, myš, perá a tablety.

### 1.2.1 Gestá

Špecifickou formou interakcie vo VR sú gestá, ktoré sú prirodzenejšie na interakciu ako využitie hardvérových ovládačov. Ide najmä o konkrétne polohu rúk ako je päšť, rôzne kombinácie vystretých prstov alebo aj následný pohyb rúk v tejto polohe. Tieto gestá je možné odchytiť a na základe toho, ktoré gesto používateľ použil, vykonať nejakú akciu, napríklad vytvorenie objektu či pohyb po prostredí. Medzi základné interakcie patrí kolízia, uchopenie, stlačenie a pustenie. [8, 9].

Existuje mnoho odvetví uplatnenia gest vo virtuálnej realite napríklad vo videohrách, vzdelávaní, marketingu, ale napríklad aj v zdravotníctve, kde môže byť používané pri trénoch operácií. Samotné zaznamenávanie rúk môže byť v blízkej budúcnosti použité aj na operácie na diaľku [2].

### Zaznamenanie gest

Na zaznamenávanie gest poznáme viacero spôsobov. Najjednoduchšie a najmenej náročné je použitie kamier priamo vo VR alebo externej kamery, ktorá sa sústreďí priamo na ruky.

Výrobcovia zariadení VR integrujú čoraz viac senzorov, ktoré umožňujú sledovanie rúk alebo očí bez akéhokoľvek ďalšieho potrebného vybavenia. Toto umožňuje široké uplatnenie v oblasti interakcie so snímanými rukami, vrátane zaznamenávania gest, kde nie je potrebný žiadny ďalší hardvér [10].

**LeapMotion** je malé zariadenie na snímanie rúk. Je uložené pod vystretými rukami a umožňuje vykreslenie celej kostry ruky. V závislosti od scenáru použitia je možné ho použiť nie len na spresnenie a odchytenie nových gest, ale aj na zaznamenávanie rúk tam, kde VR kamery nevidia, prípadne, ak naše VR zariadenie nemá alebo nepodporuje zaznamenávanie rúk [11, 12].

Na zaznamenávanie rúk a následné vyhodnotenie gest je možné použiť aj špeciálne rukavice ako napríklad **DataGlove**. Toto umožňuje rozoznať aj komplexnejšie gestá než

len jednoduché polohy rúk a aj zistiť presnejšiu polohu. Vďaka tomu je možné vykonávať vo virtuálnej realite aj zložitejšie aktivity ako napríklad šoférovanie vozidla [13].

Odchytenie gesta nemusí byť založené len na polohe ruky ale aj na základe senzorov externého zariadenia. V tomto prípade ide zvyčajne o pohybový senzor akcelerometer a rotačný gyroskop, ktoré sú dnes súčasťou aj bežných mobilných zariadení. Za gesto a možnosť interakcie tak môžeme považovať aj pohyb a polohu takéhoto zariadení čo sa dá využiť aj priamo na ovládačoch VR alebo externých ako je Wii<sup>1</sup> [14].

## **Rozpoznenie gesta**

Na získanie konkrétneho gesta zo zaznamenaných dát sa používajú jednoduchšie rovnice na základe modelu kostry zloženej z prepojených kĺbov rúk. Ďalším prístupom je vyhodnotenie gesta na základe obrázkov. Na presnejšie vyhodnotenie komplexných gest je možné použiť aj náročnejšie algoritmy ako je napríklad hlboké učenie pomocou neurónových sietí[15].

### **1.2.2 Klávesnica**

Aj vo virtuálnej realite je stále možné používať rovnaké metódy interakcie ako s počítačom. Najväčší prínos tu pridáva klávesnica, ktorá rovnako ako pri počítačoch, umožňuje písať aj vo VR. Na rozdiel od čistej samotnej vizualizácie je možné zobrazené údaje napríklad na viacerých obrazovkách meniť a pracovať rovnako ako keby sme mali doma zakúpených viacero monitorov.

Prepojenie klávesnice s VR je možné buď cez počítač alebo napriamo pre niektoré konkrétne modely klávesníc značky Logitech a Apple Magic [16].

### **1.2.3 Pero**

Kreslenie a písanie vo VR umožňuje aj špecializovaný hardvér. Jedným tipom takéhoto hardvéru sú VR perá. Wacom VR Pen[17] je pero pracujúce s tlakovými senzormi a systémom sledovania v priestore. Pomocou senzorov vie pero upravovať hrúbku písma podľa prítlaku. Ponúka možnosť využitia s grafickými tabletmi na zvýšenie presnosti písania 2D alebo 3D textov, či obrazov. Taktiež pero poskytuje kompatibilitu s veľkým množstvom VR okuliarov.

### **1.2.4 Tablet**

Tablet prepojený s virtuálnou realitou umožňuje ďalšie interakcie a to vďaka dotykom na tablete. Môžeme na jeho miesto zobraziť tlačidlá a na ne naviazať jednotlivé akcie. Podľa polohy dotyku na tablete tak vieme, ktorá akcia sa má vykonať.

---

<sup>1</sup><https://www.nintendo.co.uk/Wii>

Ďalším využitím je možnosť kresliť a písať. Doteraz sme mohli písať vo VR vďaka klávesnici alebo peru, ale chýbala nám prirodzené písanie rukou na 2D papier, alebo maľba na plátno. S týmto problémom sa stretávame často pri virtuálnych učebniciach a pracovniciach, nakoľko množstvo ľudí nie je zvyknutých písať v 3D priestore a napríklad ručné poznámky sa tak stávajú nečitateľné.

Maľbu pomocou tabletu je možné vykonávať dotykmi po tablete a následne premietat do virtuálnej reality, kde sa 2D maľba môže spojiť aj s 3D obrazom [18].

### **1.2.5 Grafický tablet**

Špeciálnym prípadom tabletu je grafický tablet. Na rozdiel od obyčajných tabletov poskytuje zvýšenú presnosť a citlivosť čo umožní aj už spomínané zapísanie čitateľných poznámok.

## **1.3 Možnosti prostredia na vývoj VR aplikácie**

Na vývoj aplikácie vo virtuálnej realite je na výber viacero grafických prostredí. Najznámejšie a najviac využívané pre prácu s virtuálnou realitou sú UnrealEngine, CryEngine a UnityEngine.

### **1.3.1 Unreal Engine**

UnrealEngine je herný engine široko využívaný na vývoj hier, simulácií, virtuálnej a rozšírenej reality pre viaceré platformy ako je počítač, herné konzoly, ale aj VR zariadenia. Poskytuje veľa nástrojov a pluginov na vývoj a podporu rôznych VR zariadení ako HTC Vice, OculusRift a Playstation VR.

Jednu z výhod je veľmi vysoká kvalita a realistickosť prostredia, objektov a efektov. Poskytuje veľkú úroveň realistického osvetlenia a vykresľovania, vďaka ktorému vývojár môže vytvárať detailné virtuálne svety. Taktiež môže vytvárať interaktívne prvky a fyzikálne javy. UnrealEngine má veľkú komunitu vývojárov, ktorá vytvára a zdieľa rôzne projekty a nástroje pre virtuálnu realitu.

### **1.3.2 CryEngine**

CryEngine je herný engine zriedkavejšie používaný na tvorbu aplikácií vo virtuálnej a rozšírenej realite. Podobne ako UnrealEngine sa zameriava na väčšinu aktuálnych platforiem. Jeho hlavnou výhodou oproti konkurencii je pomer vizuálnej kvality a výkonu. Vďaka vysoko realistickému renderovaniu a osvetleniu vie vývojár vytvárať VR prostredia z takmer reálneho sveta. Taktiež ponúka pokročilé fyzické efekty a animácie, ktoré toto prostredie o to viac oživujú. CryEngine sa zameriava hlavne na vytváranie multiplayer

hier s dynamickými scénami ako sú jaskyne, hory, rieky či celé budovy alebo mestá.

### 1.3.3 Unity Engine

Unity je aktuálne jedným z najviac používaných a odporúčaných herných vývojových prostredí na svete a je vyvíjaný rovnomenou spoločnosťou Unity Technologies. Tento engine je zostrojený tak, aby na ňom bol možný vývoj hier a aplikácií pre väčšinu platforiem ako sú osobné počítače, herné konzoly (playstation, xbox...), mobilné zariadenia, zariadenia virtuálnej či zmiešanej reality a ďalšie platformy [19].

Jednou z mnohých výhod používania Unity je množstvo nástrojov a funkcií, ktoré ponúka vývojárom na ulahčenie a zefektívnenie práce. Mnoho z nich je integrovaných priamo v Unity, ale väčšina sa nachádza v Asset store.

Taktiež Unity podporuje využitie mnohých programovacích jazykov ako je C#(C Sharp), JavaScript či Boo. Programátori môžu využívať tieto programovacie jazyky na písanie kódu, pomocou ktorého sa správa aplikácia, jej animácie, fyzika a ďalšie prvky.

Ďalšou výhodou Unity je možnosť využitia mnohých zvukových a grafických nástrojov. Unity poskytuje vlastné nástroje na úpravu a tvorbu 2D/3D grafiky, ktorými je možné modelovať rozličné typy objektov, textúr, svetla a ďalších vizuálnych prvkov. Taktiež Unity umožňuje aj použitie externých softvérov a aplikácií ako je Blender či Photoshop na tvorbu grafických a animovaných efektov.

### Asset Store

Asset store<sup>2</sup> môžeme považovať za online trhovisko s digitálnymi rozšíreniami pre Unity. Tieto rozšírenia sa nazývajú assety a zahŕňajú modely, audio nahrávky, animácie, skripty, pluginy a mnohé ďalšie. Assety môžu byť platené, ale aj voľne dostupné a pomáhajú menej zručným vývojárom pristupovať k vysokokvalitným zdrojovým kódom. Asset store je aj priamo integrovaný v Unity editore, takže zvolené assety je možno hneď importovať do projektu a využívať ich bez potreby inštalácie softvérov z tretej strany.

### Unity Hub

Dôležitou a nevyhnutnou súčasťou Unity je Unity Hub. Aplikácia Unity Hub slúži na manažment viacerých rôznych verzií Unity a projektov v nich tvorených. Pomocou tejto aplikácie môže používateľ sťahovať a inštalovať viaceré verzie Unity naraz. Vďaka tomu je používateľ schopný pracovať na projektoch, ktoré často sú náchylné na konkrétnu verziu editora. Taktiež používateľ môže do jednotlivých editorov inštalovať moduly. Moduly sú rozšírenia, ktoré obohacujú vlastnosti a možnosti editora ako je buildovanie aplikácií na

---

<sup>2</sup><https://assetstore.unity.com>

rôzne platformy, dokumentácie, jazyky, či podpory kódovania.

## **Kompatibilita Unity s virtuálnym prostredím**

Unity poskytuje vysokú kompatibilitu s VR prvkami, má integrovanú podporu pre VR a AR (rozšírená realita) prostredia, čo umožňuje vyvíjať aplikácie a hry pre tieto technológie. Taktiež poskytuje vývojárovi integrovaný obchod s platenými a neplatenými rozšíreniami, či pluginami na tvorbu a správu VR aplikácií.

Poskytuje intuitívne rozhranie a aj menej zdatnejší vývojári môžu vyvíjať aplikácie bez potreby rozsiahlych skúseností s programovaním. Taktiež má Unity množstvo integrovaných nástrojov pre tvorbu realistických a podrobných virtuálnych miestností na vysokej grafickej úrovni, pomocou ktorých používateľa vtiahne do virtuálneho sveta.

Okrem toho prostredie umožňuje vývojárom testovať a doladovať VR aplikácie priamo v Unity editore, čím zefektívňuje samotný vývoj a testovanie chýb projektu. Taktiež Unity poskytuje podporu pre väčšinu VR zariadení ako je Oculus Rift/Quest, HTC Vive, Windows Mixed Reality, čím poskytuje širokú kompatibilitu so zariadeniami.

V neposlednom rade Unity sa aktívne optimalizuje a rozširuje. Vývojári sa stále snažia zlepšovať optimalizáciu, výkon a aj stabilitu Unity engine, čo má za príčinu rýchlejších a stabilnejších VR aplikácií ako aplikácie vytvorené v inom engine. Spolu s aktívnou a rozsiahlou komunitou, ktorá sa zaoberá mnohými smermi vývoja VR, poskytuje Unity svojim nadšencom aj podrobnú dokumentáciu, návody a aj technickú podporu pri práci s Unity engineom.



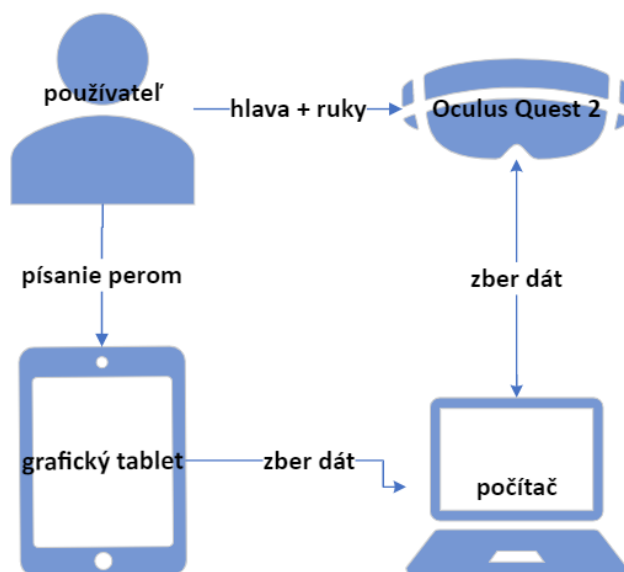
## 2 Návrh a príprava riešenia

V nasledujúcich častiach sa budeme venovať návrhu nášho riešenia, a spôsobom využitia riešenia v reálnom živote. Taktiež sa zameriame na prípravu prostredia pre našu prácu.

### 2.1 Návrh riešenia

V našej práci sa zameriame na vytvorenie systému virtuálnej pracovne, ktorý bude zaznamenávať a vyhodnocovať dáta z grafického tabletu a následne vykreslovať vstupy používateľa vo virtuálnom prostredí na plátne. Virtuálna pracovňa ponúkne viacero funkcií na modifikovanie písania ako je farba písma, mazanie, ukladanie a škálovanie plátna, či písanie prstom. Analyzujeme aktuálne možnosti a riešenia danej oblasti, ktoré aplikujeme na vylepšenie našej práce. Využijeme dostupné prostriedky prostredia Unity a jeho komunitné rozšírenia.

Na obrázku číslo 1 vidíme názorné zapojenie systému. Používateľ bude mať nasadené okuliare virtuálnej reality a zároveň pred ním bude zapojený grafický tablet s perom. Používateľ bude vo virtuálnej realite pred sebou vidieť 2D plátno a panel 3D objektov slúžiaci na interakciu so samotným písaním. Taktiež VR okuliare budú sledovať ruky používateľa. Z reálnych sledovaných rúk sa vygenerujú totožné virtuálne ruky. Používateľ uchopí pero a začne písať na grafický tablet. Dáta z grafického tabletu a pera, hlavy a rúk z VR zariadenia bude zberať pripojený počítač. Dáta sa analyzujú, spracujú a aplikujú ako výstup písma na plátne vo virtuálnom prostredí.



Obr. 1: Návrh zapojenia systému

Vďaka nášmu návrhu si používateľ bude vedieť vytvoriť virtuálnu pracovňu, v ktorej pomocou grafického tabletu a VR zariadenia bude písať texty, grafy, či diagramy rovnako ako keby písal na papier či plátno v reálnom živote. Naše riešenie bude poskytovať zvýšenú možnosť zdieľania v reálnom čase oproti papieru či plátnu. Taktiež naše riešenie môže pôsobiť ako interaktívne rozšírenie už existujúcich VR pracovní a učební, alebo využitie nášho riešenia ako základ na vybudovanie úplne novej pracovne.

Na začiatku implementácie sa zameriame na vybrané spôsoby prepojenia grafického tabletu s Oculus Quest 2 VR zariadením a počítačom. Otestujeme ich kompatibilitu a možnosti využitia pre našu prácu. Ďalej sa oboznámime s inštaláciou a konfiguráciou prostredia Unity. Vytvoríme testovacie projekty, pomocou ktorých vyvineme program na písanie vo virtuálnej realite pomocou grafického tabletu a mnoho ďalších funkcií.

Ako prvé overíme funkčnosť prepojenia grafického tabletu s VR zariadením pomocou kábla a následne pomocou Bluetooth pripojenia. Ďalšiu možnosť prepojenia otestujeme pripojením grafického tabletu do počítača. Overovanie funkčnosti a kompatibility prepojenia zistíme na základe výstupov testovacieho programu, ktorý vytvoríme v prostredí Unity a podľa potrieb nainštalujeme do VR zariadenia alebo do počítača. Z testovania si určíme najvhodnejšie zapojenie a pomocou neho budeme naďalej vyvíjať program.

Následnou nevyhnutnou časťou našej práce je zber dát, ich následná analýza a aplikovanie v reálnom čase vo virtuálnom prostredí. Dáta budeme zbierať z troch zdrojov pomocou komponentov a rozšírení v Unity. Prvý zdroj dát je samotný grafický tablet a pero. Z nich získame dáta ako je poloha a uhol pera voči tabletu, prítlak pera a mnoho ďalších dát v reálnom čase. Tieto dáta využijeme na generovanie písma na plátno a následné modifikácie. Druhý zdroj dát sú VR okuliare. Pomocou nich získame dáta ako je poloha hlavy používateľa v priestore či uhol pohľadu. Tretím zdrojom dát sú snímané ruky používateľa. Ruky budeme snímať pomocou VR okuliarov. Následne budeme generovať virtuálnu kosť a ruku vo VR prostredí na pokročilé funkcie ako sú vlasné gestá a písanie prstom po plátno.

Ako ďalší krok našej práce je analyzovanie možností písania 2D čiar v 3D virtuálnom prostredí. Vyberieme si niekoľko funkcií, ktoré ponúka prostredie Unity, ako je Trail Renderer, Paint Brush Tool či Line Renderer. Vyhodnotíme najvhodnejšiu variantu a aplikujeme ju v našej práci.

Následný krok našej práce je vytvorenie dodatočných funkcií pre zvýšenie interakcie používateľa s prostredím. Umožníme používateľovi písať na plátno pomocou prsta na virtuálnej ruke. Pomocou vlastných vytvorených gest budeme modifikovať umiestnenie plátna či škálovanie plátna. Vytvoríme vo virtuálnom prostredí stenu s interaktívnymi

objektami, pomocou ktorých budeme môcť spúšťať funkcie ako je zmena farby písma, mazanie a ukladanie plátna, či prepínanie písania medzi grafickým tabletom a prstom.

## **2.2 Možnosti prepojenia VR zariadenia s grafickým tabletom a počítačom**

Ako prvá a najmenej náročná možnosť bolo priame prepojenie VR zariadenia s grafickým tabletom pomocou USB pripojenia a Bluetoothu. V tomto prípade by používateľ nepotreboval na využívanie aplikácie dodatočný hardvér, len grafický tablet s okuliarmi Oculus Quest 2. Vytvorili sme jednoduchú aplikáciu v prostredí Unity na základnú detekciu údajov z grafického tabletu. Túto aplikáciu sme vybuildovali pre android platformy, keďže testovacie VR okuliare pracujú na báze Android.

Na inštaláciu aplikácie v samotných okuliároch sme použili aplikáciu SideQuest. SideQuest je aplikácia na manažment dát a programov pre standalone zariadenia na virtuálnu realitu ako sú Samsung Gear, Oculus Go, Oculus Quest 1 a 2. Jeho hlavnou úlohou je inštalácia a zriaďovanie aplikácií a hier mimo oficiálnych obchodov a služieb. Vďaka tomu sa SideQuest stal nevyhnutnou súčasťou samostatných vývojárov, ktorí programujú sami.

Druhá možnosť prepojenia VR zariadenia s grafickým tabletom je náročnejšia na hardvér používateľa. Táto možnosť zahŕňa pripojenie zariadení do počítača a využitie počítača ako médium. V tomto prípade sme upravili testovaciu aplikáciu z predošlej možnosti tak, aby sa buildovala pre platformu Windows operačného systému.

### **2.2.1 Prepojenie tablet s VR zariadením pomocou kábla**

Najprv sme testovali pripojenie tabletu s VR okuliarmi pomocou dátového kábla s koncovkami USB-C na oboch koncoch, ako je znázornené na obr. 2. Pripojili sme okuliare a tablet so sebou a obe zariadenia spustili. VR okuliare vypísali požiadavku o pripojenie neznámeho zariadenia, ktoré sme schválili odkliknutím na tlačidlo s názvom Accept. Tablet začal indikovať komunikáciu pomocou zelenej diódy umiestnenej na druhej strane tabletu ako sú tlačidlá. Skúšali sme v hlavnom menu VR okuliarov interagovať s tabletom, či sa neprejavia zmeny. Prostredie VR okuliarov nereagovalo na vstupy z tabletu a ani zo stláčania možných tlačidiel na tablete. Následne sme pustili testovaciu aplikáciu na detekciu vstupov z tabletu. Test sme zopakovali a výsledky boli rovnaké ako pri prvom teste, teda negatívne a žiadny výstup z tabletu sme nezaznamenali. Následne sme odpojili tablet od VR zariadenia.



Obr. 2: Prepojenie zariadení pomocou USB-C

### 2.2.2 Prepojenie tablet s VR zariadením pomocou Bluetooth

Ako druhú skupinu testovania sme si zvolili pripojenie tabletu a VR zariadenia pomocou Bluetooth pripojenia, ako je znázornené na obr. 3.



Obr. 3: Prepojenie zariadení pomocou pripojenia Bluetooth

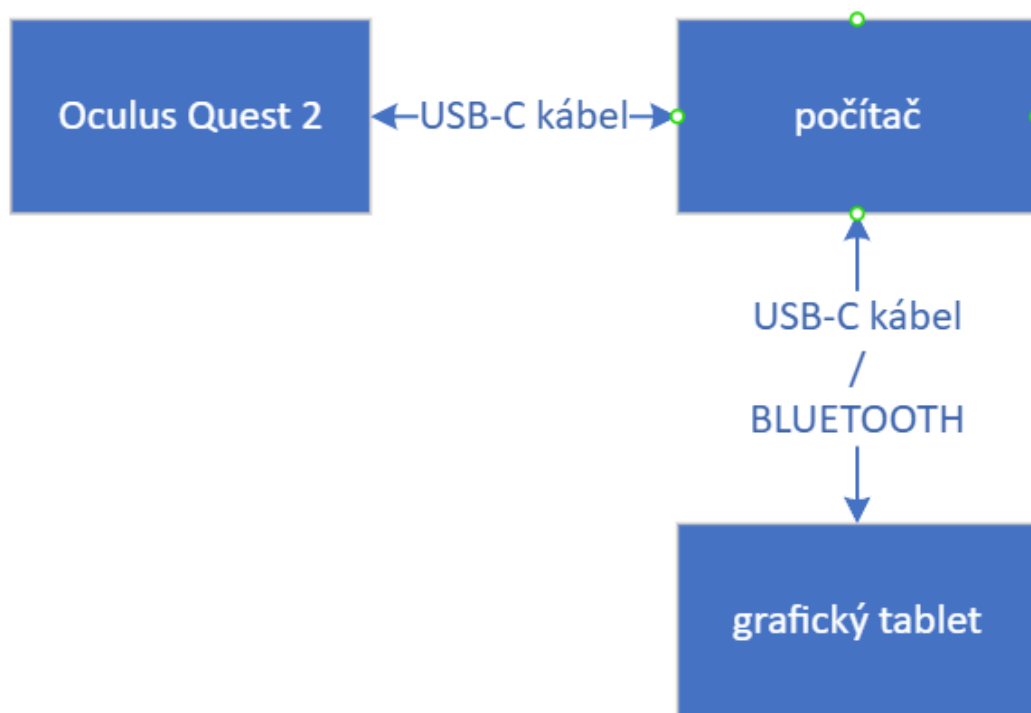
Opäť sme spustili VR okuliare a grafický tablet. Na tablete sme zapli párovací Bluetooth mód podržaním tlačidla na zapnutie spolu so stredovým tlačidlom tabletu. Po pár sekundách sa dióda umiestnená na druhej strane tabletu rozsvietila a začala blikať pravidelným intervalom modrou farbou. Následne sme vo VR zariadení v hlavnom menu zvolili možnosť kliknutím do ľavej časti panelu. Zobrazilo sa nám nové okno, kde sme otvorili nastavenia kliknutím do pravého horného rohu na ikonu ozubeného kola znázorňujúci nastavenia. Ďalšie okno s nastaveniami sa nám otvorilo s mnohými možnosťami na výber. Zvolili sme si okno s nápisom „Devices“ zariadenia a zapli sme na VR zariadení zdieľanie pomocou Bluetooth. Následne sme spustili funkciu vyhľadávania a párovania dostupných zariadení v okolí. V ponuke zariadení sme vyhládali dve zariadenia a to „BL IntuosPro L“ a „VT IntuosPro L“, čo obe tieto zariadenia predstavovali grafický tablet. Postupne sme sa párovali s oboma zariadeniami a opakovali sme testovanie ako v predošlom pokuse a to skúšať vstupy z tabletu v hlavnom menu a následne v detegovacej aplikácii.

Výsledok testov bol ten, že nie je možné zapojenie grafického tabletu s Oculus Quest 2 zariadením ani pomocou USB kábla a ani pomocou pripojenia cez Bluetooth. Po hlbšom preskúmaní našich výsledkov s výsledkami na oficiálnych fórach sme prišli k

jednotnému záveru. Kompatibilita VR zariadenia Oculus Quest 2 je obmedzovaná novšími a novšími aktualizáciami, pretože výrobca chce zamedziť spájanie nevirtuálnych pomôcok so zariadením. V aktuálnej dobe výrobca obmedzil zariadenia na len vybrané kusy klávesníc, myši a slúchadiel. Je vysoko pravdepodobné, že novšími aktualizáciami sa kompatibilita grafického tabletu navráti a bude možné tento projekt uskutočňovať bez potreby zapojenia osobného počítača do schémy.

### 2.2.3 Prepojenie tablet s VR zariadením pomocou počítača

Ako posledné sme skúšali kompatibilitu zariadení v zapojení spolu s osobným počítačom. VR zariadenie sme USB dátovým káblom zapojili do počítača, následne ďalším USB káblom sme zapojili grafický tablet do počítača, ako je znázornené na obr. 4.



Obr. 4: Prepojenie zariadení pomocou počítača

Toto zapojenie by bolo možné uskutočniť aj pomocou pripojenia Bluetooth, namiesto použitia káblov, ale nedisponovali sme dostatočne výkonným počítačom, ktorý by ponúkal Bluetooth vysielateľ a prijímač.

Vytvorili sme novú testovaciu aplikáciu, ktorú sme spúšťali v Unity editore. Táto aplikácia obsahovala základné výstupné dáta, ktoré sme vedeli zobrazit z integrovanej knižnice prostredia. Inštalácia grafického tabletu a VR zariadenia je podrobne popísaná v nasledovnej kapitole, spolu s postupom, ako sme zapojili a sfunkčnili zapojenie.

Testy zo zapojenia a spúšťania aplikácie nám preukázali funkčnosť spojenia tabletu, počítača a okuliarov. Integrovaná knižnica nám dostačovala na základný zber dát. Tento spôsob zapojenia sme si zvolili ako najspoľahlivejší a najlepší pre naše potreby vývoja VR aplikácie.

### Minimálne požiadavky počítača

Pre prácu s VR zariadeniami je nutné, aby počítač spĺňal minimálne požiadavky na chod. V našom prípade na beh zariadenia Oculus Quest 2 sú požiadavky zobazené v tabuľke 1. Špecifikácia, ktorú sme dosiahli sú zobrazené v tabuľke 2.

Tabuľka 1: Minimálne požiadavky na spustenie Oculus Quest 2

Operačný systém	Windows 10
Procesor	Intel i5-4590 / AMD Ryzen 5 1500X alebo vyššie
Pamäť	8 GB RAM alebo viac
Grafická karta	NVIDIA GTX 1060 / AMD Radeon RX 480 alebo vyššie
USB porty	Jeden USB 3.0 alebo novší
úložisko	Aspoň 5GB

Tabuľka 2: Špecifikácia na počítači, kde sa projekt vyvíjal

Operačný systém	Windows 11
Procesor	AMD Ryzen 5 3600X 6-Core Processor
Pamäť	32GB RAM
Grafická karta	NVIDIA GeForce RTX 2060
USB porty	USB 3.2 porty
Úložisko	1TB

## 2.3 Inštalácie zariadení

V tejto podkapitole sme sa venovali prvotnej inštalácií grafického tabletu Wacom IntuosPro L a VR zariadenia Oculus Quest 2. Následne sme zariadenia konfigurovali a pripravovali na použitie v projekte.

### 2.3.1 Inštalácia grafického tabletu na počítač

Na nainštalovanie grafického tabletu Wacom IntuosPro L sme potrebovali USB kábel a prístup na internet. Ako prvé sme stiahli najnovšie ovládače pre náš Wacom IntuosPro

L z oficiálnej stránky: <https://www.wacom.com/en-us/support/product-support/drivers>. Následne sme pripojili grafický tablet do počítača pomocou USB kábla a spustili sme už vopred stiahnuté ovládače. Po nainštalovaní ovládačov sme v aplikácii Wacom nastavili orientáciu tabletu a nakonfigurovali sme si tlačidlá podľa potreby. Pre riadne fungovanie pera v operačnom systéme Windows je nutné zapnutie a povolenie funkcie Pen a Windows Ink. Tieto funkcie vieme nájsť v nastaveniach Windowsu pod zložkou Bluetooth a zariadenia. V poslednom kroku sme si nastavili oblasť písania tabletu, na ktorom bude písať v počítači.

### **2.3.2 Inštalácia Oculus Quest 2 na počítač**

Na inštaláciu VR zariadenia Oculus Quest 2 sme potrebovali kompatibilný USB kábel typu USB 3.0 a vyššie. Ako prvé sme si museli stiahnuť oficiálny softvér s ovládačmi. Tento softvér sme našli a stiahli zo stránky „<https://www.oculus.com/setup/>“ pod kolónkou „Download Software“. Stiahnutý softvér sme spustili a podľa pokynov na monitore sme nainštalovali softvér s názvom Oculus do počítača. Pre zabezpečenie správneho chodu akéhokoľvek VR zariadenia je nutné mať aktuálne ovládače na grafickú kartu, ktorú máme v počítači.

Ako ďalší krok bolo vytvorenie konta Oculus alebo Facebook. Pre jednoduchosť riešenia sme si zvolili už existujúci Facebook účet. Následne sme sa prihlásili týmto kontom do aplikácie Oculus. Po prihlásení do aplikácie sme potrebovali spustiť takzvaný developerský mód. Tento mód potrebujeme na pokročilé úpravy VR zariadenia. Na zapnutie módu sa potrebujeme prihlásiť do existujúcej organizácie alebo vytvoriť novú. My sme si vybrali vytvorenie novej organizácie. Vyplnili sme formulár na oficiálnej webovej stránke Meta <https://developer.oculus.com/manage/organizations/create/>. Po vyplnení formuláru sme museli potvrdiť účet prihlásením na stránke: <https://developer.oculus.com/manage/verify/>. Stránka od nás vyžiada jedno z nasledovných a to potvrdenie nášho telefónneho čísla a nastavenie SMS dvoj-faktorovej autentifikácie alebo pridanie platobnej metódy na našom účte.

Po nainštalovaní softvéru a vytvorení konta s developerskými prístupmi sme spustili Oculus Quest 2. Spustil sa nám sprievodca prvého spustenia zariadenia. Postupovali sme podľa pokynov až po koniec návodu, kde sme sa prihlásili pod kontom, ktoré sme si predtým zvolili. V rýchlych nastaveniach, umiestnených v ľavom spodnom rohu sme si zvolili možnosť „Settings“. Zobrazilo sa nám okno s kolónkami a zvolili sme si System a následne Developer. V okne sme si zaškrtnuli možnosť USB Connection Dialog. V aplikácii Oculus sme v nastaveniach povolili aplikácie od neznámych zdrojov, verejný testovací kanál a runtime funkcie pre vývojárov. Následne sme zapojili VR okuliare k počítaču pomocou

kábla. Vo VR okuliároch sa nám zobrazilo okno na povolenie pripojenia zariadenia, ktorý sme potvrdili.

Po týchto krokoch si otvoríme rýchle nastavenia vo VR okuliároch a zobrazí sa nám okno na pravej strane s nápisom Quest Link. Stlačíme ho a po načítaní sa ocitneme v šedom mriežkovanom priestore a pred nami sa zobrazí panel možností. V pravej časti zvolíme monitor a zobrazí sa nám displej počítača a vieme ho ovládať pomocou kontrolérov.

## **2.4 Inštalácia prostredia a potrebných doplnkov**

V nasledujúcich podkapitolách sme sa venovali inštalácií a nastavovaní prvotných konfigurácií prostredia Unity, Unity editora a kompilátoru Visual Studio 2019.

### **2.4.1 Inštalácia Unity hub**

Pre nainštalovanie Unity Hub musíme navštíviť oficiálnu stránku Unity „<https://unity.com/>“ kde stlačíme tlačidlo „Download Unity hub“ v pravej hornej časti stránky. Stiahneme dostupný inštalačný balíček na nami zvolený operačný systém, v tomto prípade Windows OS. Stiahnutý balíček nainštalujeme podľa pokynov v sprievodcovi inštalácie. Následne si musíme vytvoriť konto, pod ktorým budeme pracovať. Vďaka školskému kontu G-Suite si môžeme vytvoriť konto so školskou licenciou na Unity.

### **2.4.2 Inštalácia Unity editora**

Po jednoduchej inštalácii Unity Hub a prihlásení konta si môžeme vytvoriť Unity editor, v ktorom budeme naďalej pracovať. V ľavej časti aplikácie si zvolíme možnosť Installs a stlačíme tlačidlo Install Editor. Zobrazí sa nám okno s ponukou dostupných editorov. Zvolíme si oficiálne vydania a nainštalujeme najnovší editor s označením LTS. Táto možnosť je taktiež označená ako odporúčaná a zabezpečí nám dlhodobú podporu editora. V čase práce bola zvolená verzia 2021.3.15.f1 LTS. Zobrazí sa nám okno s možnosťami doinštalovania dodatočných modulov do editora. Zvolili sme si moduly inštalácie Microsoft Visual Studio Community 2019 spolu s podporou pre Android platformy spolu s OpenJDK a Android SDK & NDK Tools, keďže Oculus okuliare pracujú na báze Androidu. Taktiež sme označili inštaláciu Windows Build Support (IL2CPP) pre testovanie aplikácie na počítačoch s Windows operačným systémom. Následne sme editor dali inštalovať. Na konci inštalácie sa nám spustila druhá inštalácia a to programu Visual Studio Community 2019.

### **2.4.3 Inštalácia Visual Studio**

Pri inštalácii Visual Studio sme opäť vyberali dodatočné modifikácie programu. Pre plynulé a kompatibilné programovanie sme si zvolili .NET desktop development, Desktop



development with C++, Game development with Unity, Game development with C++ a individuálny komponent Windows 10 SDK (10.0.16299.0).

Po inštalácii Unity Hubu, Unity editora a programu Visual Studio sme reštartovali počítač pre správny chod programu. Následne spustíme Unity Hub a sme pripravení na vytváranie projektov a začať pracovať s Unity.

## **2.5 Spojenie zmiešanej reality s virtuálnou**

Využitie zmiešanej reality (AR) v prostredí VR pridáva o stupeň vyššiu interakciu s prostredím a navyšuje zážitok používateľa. Hlavným zmyslom využívania je aplikovanie reálnych objektov do virtuálnej reality pomocou špeciálnych kamier, markerov a softvéru. Odkúšali sme dve možnosti spojenia AR s VR prostredím pomocou integrovaných kamier VR a pomocou triangulácie s dodatočnou kamerou.

### **2.5.1 Vuforia Engine**

Na prepojenie zmiešanej a virtuálnej reality sme v projekte využili asset Vuforia Engine. Vuforia Engine je jednou z mnohých platforiem pre AR prostredia. Zameriava sa na mobilné zariadenia a smartglasses. Špecializuje sa na sledovanie AR markerov, detekciu pohybu, analyzovanie obrazu a ďalšie vymoženosti. Vďaka jej knižniciam kompatibilných s programovacím jazykom C# vie spolupracovať s prostredím Unity.

Pre riadne používanie Vuforia Engine sme museli vytvoriť konto na oficiálnej webovej stránke programu. Po úspešnej registrácii konta sme sa opätovne prihlásili na oficiálnej stránke. V zložke Develop a License Manager sme požiadali o Basic licenciu, kde sme zadali aj názov pre našu licenciu. Basic licencia poskytuje dostatočné možnosti pre náš projekt. Vytvoril sa nám licenčný kľúč, ktorý budeme potrebovať neskôr.

V našom projekte sme sa pokúšali o vytvorenie AR markerov. Tieto markery by snímali integrované kamery Oculus Quest 2 okuliarov a generovali by tablet vo VR prostredí, ktorý by odpovedal reálnemu grafickému tabletu. Vuforia Engine sme stiahli pomocou integrovaného Asset storu v Unity a importovali sme to do nášho projektu. Jedná sa o voľne dostupnú verziu programu. Následne sa nám v Unity editore pod zložkou Window objavila možnosť konfigurácie Vuforia Engine. V konfigurácii sme zadali licenčný kľúč, ktorý sme si vygenerovali v predošlých krokoch.

Ako prvý krok sme vytvorili nový testovací projekt na overenie funkčnosti Vuforia funkcií. V projekte sme vytvorili nový objekt Vuforia engine Image Target. Tento objekt slúži ako predloha, aký obraz má kamera zaznamenať, aby vytvorila AR objekt. Následne sme upravili komponenty objektu, aby vyhovoval našim potrebám. Zmenili sme typ

generujúceho objektu z databázy na obrázok. Pomenovali sme Target Name na target. Vložili sme obrázok terča do zložky Image v komponente. Následne sme vytvorili 3D objekt podobný tabletu v modelovacom programe Blender. Importovali sme daný objekt do projektu a umiestnili sme tento objekt ako detsky objekt ImageTarget. Ako poslednú úpravu sme hlavnej kamere priradili komponenty Vuforia Behaviour(Script) a Default Initialization Error Handler(Script).

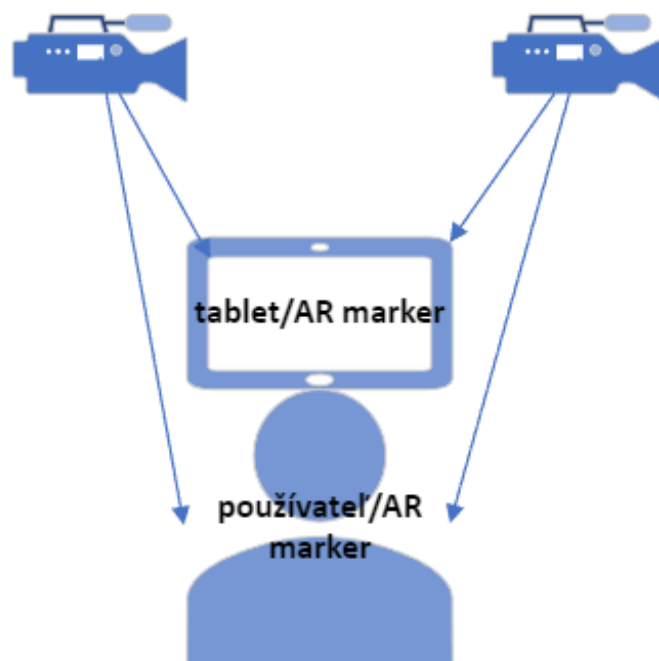
Vytlačili sme si na papier AR markery, aké sme si udali v projekte. Testovali sme dočasný projekt spolu s webovou kamerou a vytlačenými AR markermi. Vždy keď sme kameru zamerali na marker, tak sa nad ním vytvoril požadovaný objekt. Overila sa nám funkčnosť Vuforia funkcií.

Následne sme postup zopakovali na VR projekte. Na využitie predných kamier Oculus Quest 2 sme pridali komponent OVR Passthrough Layer. Pri testovaní VR projektu sa nám podarilo vytvoriť AR prostredie pomocou VR zariadenia tým, že používateľ videl reálne okolie cez predné kamery Oculus Quest 2. Pri rozsiahlejšom testovaní sa nám nepodarilo preukázať funkčnosť predných kamier spolu s AR markermi z Vuforia Engine.

Nefunkčnosť je spôsobená aktuálnou verziou VR zariadenia. Z bezpečnostného hľadiska pokročilé funkcie predných kamier Oculus Quest 2 sú nedostupné pre vývoj nelicencovaných softvérov z dôvodu zabezpečenia súkromia používateľa. V budúcnosti je možné, že táto funkcionality bude opäť funkčná a spojenie VR sveta s AR prvkami pomocou Oculus Quest 2 bude dosiahnuteľné.

### **2.5.2 Triangulácia**

Ďalšou možnosťou aplikovania AR prvkov je využitie triangulácie pomocou ďalších kamier, ako je vidieť na názornom obrázku 5. Triangulácia je proces vytvárania 3D prostredia a objektov pomocou 2D obrazov. Využíva sa v robotike, geodézii, virtuálnej realite a mnohých ďalších odvetviach. Proces triangulácie prebieha tak, že sa vypočítajú 3D súradnice z 2D obrazov kamier rôzne rozložených. Používajú sa hlavne stereo kamery alebo kamery s rovnakou orientáciou. Najprv sa tieto kamery nakalibrujú a následne sa použije algoritmus triangulácie, aby vedel vypočítať 3D súradnice požadovaných markerov.



Obr. 5: triangulácia pomocou kamier

Pre náročnosť výpočtov, nedostatku času a absencie potrebného hardvéru sme túto možnosť zamietli. Pomocou triangulácie by sme vedeli v budúcnosti aplikovať program Vuforia s Unity rozhraním a preniesť VR projekt do zmiešanej reality. Prípadne modifikovať sfunkčnené postupy a aplikovať pokročilé technológie.

## 2.6 Tvorba projektu

V tejto časti sme sa venovali prvotnému vytvoreniu projektu v prostredí Unity a následných prvých krokoch konfigurácie. Ako správne vytvoriť projekt pre naše účely a ako daný projekt spustiť v editore. Využitie integrovaného obchodu na inštaláciu potrebných rozšírení a balíčkov. Vytvorenie a úprava základných objektov v prostredí ako je zem a plátno. Importovanie a následná konfigurácia hráčovej hlavy a rúk do prostredia.

### 2.6.1 Vytvorenie nového projektu v Unity Hub

Po predchádzajúcich inštaláciách môžeme vytvoriť nový projekt. V Unity Hub v ľavej časti zvolíme možnosť Projects a vytvoríme nový projekt. Zadáme názov projektu a jeho umiestnenie na počítači. Z výberu šablón si zvolíme 3D jadro. V projektoch sa nám vytvorí nový riadok s novovytvoreným projektom. Skontrolujeme si správny názov a či je zvolená správna verzia editora. Následne spustíme projekt tým, že naň klikneme. Otvorenie nového projektu môže trvať aj niekoľko minút.

### 2.6.2 Inštalácia pluginov

V otvorenom projekte spustíme Asset Store, ktorý sa nachádza v hornej lište pod možnosťou Window. Otvorí sa nám webový prehliadač spolu s Asset Storom. Vyhľadáme a nainštalujeme rozšírenie s názvom Oculus Integration a importujeme ho do projektu. Následne v projekte otvoríme Windows>Package manager, kde máme uložené všetky dostupné assety a nainštalujeme dodatočné balíčky s názvami XR Plugin Management, OpenXR Plugin, Oculus XR Plugin a ostatné predinštalované balíčky aktualizujeme na najnovšie verzie.

### 2.6.3 Prvé kroky v projekte

Po nainštalovaní assetov a pluginov môžeme pristúpiť k prvým krokom v projekte. Ako prvé sme vytvorili 3D objekt typu plane s názvom podlaha. Objektu podlaha sme zresetovali transformáciu aby sme znulovali koordináty a škálovanie všetkých troch rozmeroch upravili na hodnotu jedna. Ďalej sme stiahli z Asset storu vhodnú textúru, v našom prípade textúru súkružníc, textúru sme pretiahli na podlahu, a tým zmenili predvolenú textúru na novú. Týmto máme vytvorený základ, na ktorom budeme stáť.

Vytvorili sme ďalší 3D objekt plane s názvom plátno. Aj tomuto objektu sme zresetovali transformácie a následne sme upravili pozície plátna, aby približne odpovedali umiestneniu stola, na ktorom pracujeme, v našom prípade na koordináty (0 ; 1,125 ; 0,4). Taktiež sme znížili škálovanie, aby približne odpovedali tabletu, s ktorým testujeme, a to na (0,043 ; 1 ; 0,029). Vytvorili sme nový biely materiál a aplikovali ho plátnu, aby bol jasne odlišný od okolia.

### 2.6.4 Importovanie hlavy hráča do projektu

Pre vytvorenie hráča najprv vytvoríme prázdny objekt v hierarchii s názvom hráč. Prázdnomu objektu hráč sme zresetovali transformáciu. Pod objekt hráč sme priradili dva prefaby s názvom OVRCameraRig a InputOVR ako deti objektu. V detskom objekte OVRCameraRig sme upravili tieto nastavenia:

Tracking Origin Type sme zmenili z Floor Level na Stage. Stage nastavenie berie do úvahy výšku používateľa pri spustení aplikácie a upraví umiestnenie hlavy v projekte podľa toho. Vďaka tomu sa výška hráča nastavuje dynamicky pre každého používateľa.

Hand Tracking Support sme zmenili na Hands Only. Hands Only nám zabezpečí plynulý vývoj a následné používanie aplikácie, pretože VR okuliare budú zaznamenávať len samotné ruky a nie kontroléry.

Hand Tracking Frequency sme zmenili na HIGH pre plynulejšie zaznamenávanie rúk,

tým že zvýšime obnovovaciu frekvenciu detekcie rúk. Pre slabšie počítačové zostavy je toto nastavenie radšej nastavené na úroveň LOW.

Hand Tracking Version sme nastavili na verziu 2. Oproti verzií jedna V2 využíva dve kamery na detailnejšie sledovanie pohybu rúk. Taktiež táto verzia poskytuje podrobnejšie rozpoznávanie gest a celkovo ponúka používateľovi prirodzenejšie prostredie virtuálnej reality s vyššou presnosťou.

### **2.6.5 Importovanie rúk hráča do projektu**

Po upravení hlavy hráča sme začali upravovať ruky. Na úpravu rúk sme najprv pridali do objektu OVRVameraRig pod detskými objektami LeftHandAnchor a RightHandAnchor prefab s názvom OVRHand a premenovali sme ich podľa detských objektov ako LeftOVRHand a RightOVRHand.

Pre LeftOVRHand a RightOVRHand upravíme nasledovne:

Nastavíme Hand Type a Skeleton Type na Hand Left pre ľavý objekt a Hand Right pre pravý objekt, aby to odpovedalo jednotlivým detským objektom. Toto nastavenie automaticky vygeneruje textúru a kosti správnych rúk.

Pre oba objekty priradíme pod Ponter Pose Root celý prázdny objekt hráč. Vďaka priradeniu môže pracovať skript OVRHand správne. Tento skript zabezpečuje sledovanie reálnych rúk používateľa a generovanie virtuálnych rúk vo VR prostredí ako aj škálovanie generovaných rúk, overovanie základných gest, generovanie a analyzovanie virtuálnej kostry podľa reálnych rúk používateľa.

## 3 Implementácia

Po vytvorení a pripravení projektu v Unity na základe predchádzajúcej kapitoly a prepojení grafického tabletu s VR zariadením, môžeme prejsť k samotnej implementácii. V našej práci sa zameriavame na vytvorenie a vývoj jednoduchej aplikácie na písanie a kresbu pomocou grafického tabletu.

Vývoj začína zberom dát z grafického tabletu, hlavy a rúk používateľa, na základe ktorých následne vykresľujeme čiary v 3D virtuálnom prostredí. Na to, aby sme umožnili viac interakcií používateľa sme pridali viacero možností úprav, ako je farba písma, zmena písacieho nástroja a modifikovanie plátna pomocou špecifických gest.

### 3.1 Zber dát

Nevyhnutnou súčasťou našej práce je zber dát používateľa na zachytenie jeho správania v našom projekte a interakcií s jednotlivými objektami vo VR prostredí. Hlavným predmetom zberu dát je získavanie súradníc a prítlaku pera pri písaní s tabletom. Okrem toho sledujeme a zaznamenávame pohyby rúk na vykreslené kostry ruky, ktoré následne využívame na pokročilé funkcie a ich spúšťanie.

Dáta zbierame na troch vrstvách a to dáta z grafického tabletu, dáta z VR zariadenia a obraz z predných kamier VR zariadenia. Následne sme tieto dáta pomocou rôznych skriptov analyzovali a spracovali. To, ako sme tieto dáta zberali a využívali, sme rozvinuli v ďalších kapitolách.

#### 3.1.1 Zber dát grafický tablet

Dáta z grafického tabletu prijímame priamo pomocou Unity. Unity má natívne integrovaný balík s názvom Input System, pod ktorý spadajú vstupy z mnohých typov vstupov ako je myš, klávesnica, kontrolér a aj grafický tablet. Na otestovanie a overenie aké dáta vieme a nevieme získať z tabletu sme vytvorili testovací skript s názvom InputTest. V tomto skripte získavame dáta ako je poloha pera z hľadiska monitora, či je pero držané tesne nad tabletom, či je pero pritlačené o tablet, tlak akým je pero pritláčané o tablet, jeho sklon voči tabletu v dvoch osiach. Tento skript sme priradili objektu plátno. Následne sme tablet pripojili k počítaču a spustili projekt. Tak ako sme písali na tablete sa nám zobrazovali dáta v reálnom čase v projekte. Týmto skriptom sme dokázali kompatibilitu grafického tabletu Wacom IntuosPro L voči počítaču a aj aké typy dát vieme pomocou tohto tabletu získať, keďže niektoré tablety poskytujú obmedzené množstvo dát.

Skript sme v ďalších postupoch upravovali tak, aby nám vyhovoval na zber konkrétnych a podstatných dát. Zamerali sme sa na pozíciu pera v čase a či je dané pero pritlačené

o tablet. Pomocou zhromažďovania a analyzovania dát v reálnom čase sme umožnili používateľovi naraz písať na grafickom tablete a generovať písmo vo virtuálnej realite.

### **3.1.2 Zber dát hlava hráča**

Dáta z VR okuliarov sme získali pomocou Oculus Integration balíčka, ktorý sme nainštalovali v predošlej kapitole „3.1.Vytvorenie nového projektu (projekt+plugíny)“. Z balíka sme konkrétne využili objekt OVRCameraRig a jeho detských objektových kamier Right/Left/Center EyeAnchor. Pomocou týchto objektov a detských objektov vieme zaznamenávať pozíciu používateľovej hlavy, kam je používateľ natočený a upraviť dynamickú výšku používateľa. Nadalej sa dáta nazbierajú z vygenerovaných objektov v priestore, ako je plátno a interaktívna stena, na možnú interakciu a textúrovanie.

Ďalší druh dát, ktoré sa v projekte využili, sú obrazy z predných kamier. Na získavanie dát sa využili komponenty ako je OVR Passthrough Layer (Script) a OVR Headset Emulator (Script) na sprístupnenie AR zmiešanej reality a analyzovanie dát pre AR prvky ako sú markery a detekcia objektov.

### **3.1.3 Zber dát ruky hráča**

Ruky sa generujú s využitím prefabu OVRHand. Dáta z rúk sa zhromažďujú, ak ich má používateľ v zornom poli. Pri vykresľovaní rúk sa taktiež generuje aj takzvaná kostra ruky. Zoznam kostí, ktoré sa generujú v každej ruke zvlášť sa nazýva List<OVRBone>. Každá kosť v tomto zozname obsahuje množstvo použiteľných dát ako sú lokálne pozície kosti, ich škálovanie, rotácia, uhol kostí a mnohé ďalšie informácie. na detekciu a analyzovanie funkcií rúk v reálnom čase.

V projekte sme využili zoznam kostí List<OVRBone> na viacero interaktívnych funkcií. Pozície kostí sme využili na detegovanie jednotlivých špeciálnych gest pomocou vzdialeností jednotlivých kostí od seba. Transformáciu ukazováka sme využila na písanie čiar po plátne. Taktiež sme využili základné integrované gestá na manipuláciu s plátnom.

## **3.2 Spôsoby písania čiar v Unity**

Unity ponúka viacero komponentov a možností ako môže používateľ generovať čiary a text v prostredí. Výhodou týchto komponentov je to, že sú už priamo integrované v základnom Unity projekte a nemusia sa doinštalovať dodatočne treťostranné softvéry alebo nakupovať ich v Asset Store. Najčastejšie možnosti na písanie v Unity sú Trail Renderer, Paint Brush Tool, UI canvas a Line Renderer.

### **3.2.1 Trail Renderer**

Trail Renderer je komponent v Unity, ktorý vytvára optické efekty za pohybujúcim sa objektom v reálnom čase. Tieto stopy môžu byť použité ako lúče svetla, efekty ohňa, vetra, jednoduchých čiara podobných efektov či javov. Využíva sa pri väčšine v hrách na vytváranie dojmu pohybu objektu ako je let šípky alebo na vytváranie animácií ako sú lúče slnka.

Trail Renderer tvorí čiary v trojrozmernom priestore, ktoré sa vedia modifikovať ako je zmena farby alebo textúry, rozširovanie a zužovanie, doba životnosti čiar a ďalšie modifikácie. Tieto modifikácie sa vedia upravovať v reálnom čase a tým umocňovať pocit realistických efektov.

Využitie tohto komponenta je časovo náročné a pri menej výkonných počítačoch či standalone VR headsetoch to môže značne ovplyvniť výkon aplikácie. Taktiež nižšia frekvencovanosť použitia a optimalizácia je podstatnou nevýhodou pri práci s písaním v Unity projekte.

### **3.2.2 Paint Brush Tool**

Paint Brush Tool je nástroj v Unity, pomocou ktorého vie používateľ kresliť rôzne 2D ale aj v 3D objekty v priestore. Tento nástroj sa hlavne využíva na dekorovanie prostredia a herných panorám. Paint Brush Tool umožňuje vkladať farebné body, čiary alebo objekty na povrchy podobne ako kreslenie v reálnom živote. Používateľovi umožňuje úpravu už existujúcich objektov alebo vytvárať úplne nové objekty od začiatku. Taktiež si používateľ môže vyberať z mnohých druhov nástrojov na maľovanie ako sú rôzne druhy štetcov, dodávanie efektov objektom ako sú svetelné efekty, tieňovanie a textúry.

Jedna z vlastností, ktorá je zriedkavá u iných kresliacich komponentov či funkcií je možnosť používania rôznych metód kreslenia. Jednou z tých metód je kreslenie pomocou voxelov, ktorý používateľovi umožňuje kreslenie 3D objektov ako keby skladal na seba farebné kocky. Taktiež podporuje viacero módov kreslenia, ako je normal mode, hue mode, saturation mode a value mode. Pomocou tých módov sa aplikovaná farba vie modifikovať viacerými spôsobmi ako je samotná zmena farby, jej sýtosť, jas a odtieň. Tieto módy vie komponent podľa potrieb zmiešať a vytvoriť realistický dojem maľovania.

### **3.2.3 UI Canvas**

UI Canvas je jedným zo základných komponentov v Unity na vytváranie užívateľského rozhrania. Slúži hlavne na vytváranie grafických objektov ako sú, obrázky, tlačidlá a texty. Všetky tieto prvky sú umiestnené na takzvanom canvase, ktorý vidí kamera používateľa.



To znamená, že kam sa používateľ alebo hlavná kamera pozerá, tak tam sa generuje canvas.

UI Canvas je možné využiť na kreslenie v kamere, ako napríklad v 2D priestore alebo pri využití dotykových obrazoviek a počítačových myšiek. V našom projekte nevieme tento komponent využiť, keďže kreslíme v 3D VR priestore na objekty a nie na samotnú kameru.

### 3.2.4 Line Renderer

LineRenderer je Unity komponent, ktorý sa používa na vykreslenie čiary na základe množiny bodov v 3D priestore. Čiara môže byť definovaná radom bodov a môže byť upravená v reálnom čase, aby sa vytvorili rôzne efekty, ako sú chodníky, laserové lúče a ďalšie. LineRenderer možno použiť aj v spojení s ďalšími funkciami Unity, ako je fyzika, skriptovanie a materiály, na vytváranie komplexnejších a interaktívnejších efektov založených na riadkoch.

Line Renderer má uložené za sebou idúce pozície, ktoré vie vývojár upravovať, pridávať alebo odoberať. Taktiež ponúka možnosti ako hrúbka čiary, vytvorenie vlastnej textúry a ďalšie úpravy ako použitie globálnych premenných.

Pre jednoduchosť a efektivitu využitia sme si pre náš projekt vybrali Unity komponent Line Renderer. V našom prípade sme Line Renderer využili ako čisto písací nástroj na kreslenie 2D čiar v 3D priestore, ktorý sme využili na dva rôzne spôsoby písania. A to na písanie pomocou grafického tabletu s perom a na písanie pomocou prsta na pravej ruke používateľa.

### Písanie pomocou pera a tabletu

Ako prvý krok pre písanie do projektu grafickým tabletom sme museli vytvoriť samotný štetec prefab a atribúty týkajúce sa štetca. V projekte sme vytvorili prázdny objekt, volaný ako brush (štetec), ktorému sme priradili komponent LineRenderer. Ten sme si upravili tak, aby nám vyhovoval pre naše použitie. Po úpravách sme následne tento objekt zmenili na predkonfigurovaný herný objekt, takzvaný prefab, ktorý vieme neskôr viackrát použiť v danej scéne.

Znížili sme hodnotu Position Width z 1 na 0.15. Vďaka úprave hodnoty sme naškalovali hrúbku písma tak, aby zhruba odpovedal škálovania celého prostredia v projekte. Naďalej budeme môcť túto hodnotu meniť podľa potrieb písania detailnejších častí písma.

Nastavili sme Position Size na 2. Na správne inicializovanie komponenta LineRenderer potrebujeme aspoň dva prvotné body. Pomocou úpravy prefabu na dva sme sa vyvarovali možným závadám pri spúšťaní projektu a pri vytváraní každej novej čiary.

Alignment sme nastavili na Transform Z. V základnom nastavení tvorba písma

funguje tak, že samotné písmo rotuje podľa natočenia hlavnej kamery používateľa. Vďaka nastavenia transformácie na zetovú os vieme zaručiť, že všetko písmo vytvorené týmto prefabom bude zrotované tak, aby bolo zarovno s plátnom.

Vytvorili sme nový základný materiál, ktorý bude slúžiť ako prvotná farba pre štetec a priradili sme tento materiál do prefabu. Naďalej sme ešte deaktivovali prijímanie tieňov, aby nám nevznikali anomálie a odľahčili systém.

Písmom bolo potrebné upraviť podľa nášho projektu, aby sa s ním vedeli náležite nárábať. Vytvorili sme skript s názvom Draw a priradili sme tento komponent objektu plátno. Na začiatku skriptu sme nastavili dynamické škálovanie prepočtu písma z grafického tabletu na písmo vo VR prostredí. V základe sú tieto hodnoty nastavené tak, aby vyhovovali rozlíšeniu testovacieho monitora na 1920x1080 pixelov a vytvorili mierku rozšírenia, podľa ktorej určujeme presnosť písania. Obe hodnoty sme označili ako verejné premenné, takže sme ich vedeli v prípade potreby upravovať priamo v prostredí Unity.

Pri každom stlačení pera o grafický tablet sa spustí druhá časť skriptu. Pomocou integrovanej knižnice vstupov skript zistí umiestnenie pera na tablete. Umiestnenie prepočíta cez preddefinované škálovanie, pričom sa vymenia koordináty Z s koordinátami Y. Táto výmena je nutná, keďže tablet odovzdáva výstupy v tvare X, Y a Unity prostredie pracuje s koordinátami X, Z a výškou Y. Následne sa vytvorí nový detský objekt z prefabu štetca. Zadájú sa mu prvé dve sady súradníc, keďže na správne spustenie Line Renderera sú nutné dva body. Tieto dva sady súradníc je aktuálne umiestnenie pera v reálnom čase.

Kým používateľ drží pero na tablete s tlakom vyšším ako nula, tak skript každý snímok kontroluje, či sa zmenila pozícia pera. Ak táto zmena nastala, tak sa vytvorí nová prázdna pozícia čiary a následne ju naplní koordinátami. Táto akcia sa opakuje každý snímok kým pero používateľ nezastaví alebo nezdvihne pero z tabletu. Ak sa pero z tabletu zdvihne, tak sa objektu prestanú vytvárať nové sady súradníc a dokončí sa vytváranie tohto detského objektu. V prípade ďalšieho priloženia pera sa vytvorí nový detský objekt a skript opakuje totožné kroky ako pri predchádzajúcom objekte.

## **Písanie pomocou prsta na ruke**

Ako prvý krok v písaní pomocou prstu vo VR prostredí sme taktiež museli vytvoriť a upraviť prefab štetca tak ako v predošlej kapitole. Jedinou zmenou oproti prefabu písania tabletom je hodnota šírky samotného štetca Positions Width na hodnotu 0,2. Na rozdiel od písania pomocou grafického tabletu sme zvolili rozdielnu metódu zaznamenávania písania s využitím takzvaných triggerov.

Trigger je komponent v Unity, ktorý slúži na detekciu kolízií medzi objektami. Ak

objekt zasahuje do druhého objektu, tak trigger pošle správu o kolízií, pomocou ktorej vieme spúšťať skripty alebo akcie. Táto kolízia pritom nemusí byť ani fyzická, to znamená, že objekty sa od seba nemusia odraziť alebo zdeformovať.

V našom projekte sme triggerery využili tak, že sme vytvorili 3D objekt gule s názvom PokePoint. Škálovanie sme upravili na hodnotu 0,03 vo všetkých rovinách. Guli sme priradili špeciálne označenie „tag“ s názvom Respawn. Umiestenie gule sme nastavili na nulové hodnoty, keďže sa táto guľa bude pohybovať.

Následne sme vytvorili ďalší 3D objekt kváder s názvom HitBox. Kváder sme nastavili ako detský objekt plátna. Zresetovali sme objektu hodnoty transformácie, aby stred kvádra odpovedal stredu plátna. Upravili sme rozmery kvádra tak, aby odpovedal rozmerom plátna, iba škálovanie v rozmere Y sme nastavili tak, aby bola tesne nad a pod plátnom. Deaktivovali sme komponent Mesh Renderer, vďaka ktorému kváder nebolo vidieť v prostredí. Ako posledné sme aktivovali objektu funkciu Is Trigger. Tým pádom máme neviditeľný objekt, ktorý bude zapínať skripty pri kolíziách s objektom označeným ako Respawn.

Vytvorili sme nový skript s názvom FingerWrite a priradili ho objektu plátno. Na začiatku skriptu sa čaká na vygenerovanie rúk používateľa. Po vytvorení pravej ruky a jej kostí skript skontroluje list kostí a vyhľadáva kosť s názvom Hand\_IndexTip, ktorý predstavuje posledný článok ukazováku na ruke, inak nazývaný ako distálna falanga. Keď kosť skript nájde, tak začne sledovať jeho pozíciu. V reálnom čase na túto pozíciu sa presúva aj objekt gule, ktorú sme vytvorili v predchádzajúcom kroku. Na to, aby skript dokázal písať pomocou prstu, sa musela spraviť korekcia generovania bodov písania. Tá sa zabezpečila pomocou matematického prepočtu lokálnych premenných plátna voči globálnym premenným.

Ak vygenerovaná guľa s označením Respawn sa pretne s kvádom HitBox, tak sa spustí druhá časť skriptu. Vytvorí sa nový detský objekt z prefabu štetca. Zadájú sa mu prvé dve sety súradníc, keďže na správne spustenie Line Renderera sú nutné dva body. Hodnoty pre prvé dva sety sú aktuálne pozície kolízie objektov.

Kým používateľ udržiava ukazovák, teda objekt PokePoint, v objekte HitBox, tak skript každý snímok deteguje, či sa nezmenila pozícia kolízie. Ak nastala zmena pozície kolízie, tak sa vytvorí nová prázdna pozícia čiary a následne ju naplní koordinátmi. Skript opakuje detekciu každý snímok, až kým sa neukončí daná kolízia. V prípade ukončenia kolízie sa tvorba objektu dokončí. Ak by nastala nová kolízia, tak sa vytvorí ďalší detský objekt a postup generovania čiar sa opakuje ako v predošlých krokoch.

### 3.3 Úpravy písma

Nevyhnutnou súčasťou písania vo virtuálnej miestnosti sú možnosti ako upravovať jednotlivé parametre písma. Úpravy sme rozdelili do viacerých kategórií, aby používateľ mal prirodzenejší výber a to zmeny farby písma, premazanie a ukladanie plátna, prepínanie písma medzi grafickým tabletom a prstom, úpravy pozície plátna a výšky plátna. Základom takýchto úprav je samotné ovládanie nastavenia písma. Využili sme technológie triggeru ako pri písaní pomocou prsta a vytvorili jednoduchú stenu s možnosťami modifikácie písania. Na stene umiestnenej po ľavej strane používateľa sme vytvorili viacero guľových 3D objektov na úpravu písma. Vybrali sme si ľavú stranu, pretože predpokladáme, že majorita možných používateľov budú praváci, to znamená, že ovládanie ľavou rukou bude prirodzenejšie. Každý objekt gule má v sebe uložený skript, ktorý sa spustí pri kolízii s akýmkoľvek objektom.

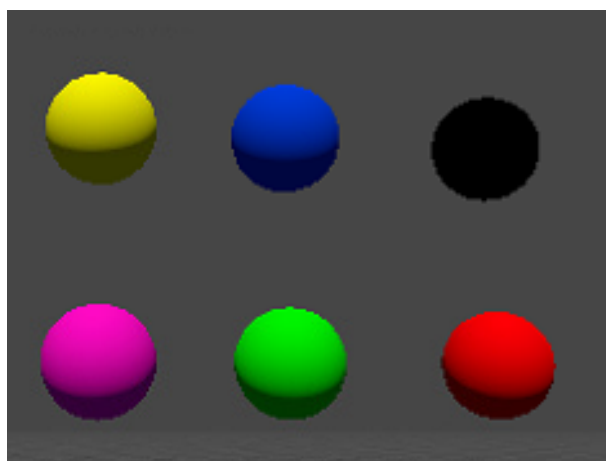
#### 3.3.1 Úpravy písania - farba

Prvý štýl upravovania písania je zmena farby písma, je to najjednoduchší typ úpravy. Vytvorili sme šesť nových materiálov s farbami modrá, zelená, čierna, červená, fialová a žltá ako je vidieť na obrázku 6. Materiály sme aplikovali na šesť rozličných objektov gúľ, ktoré sme umiestnili vedľa seba. Každému objektu gule sme aktivovali parameter is Trigger, aby sme vedeli spúšťať skripty založené na kolízií s týmito objektami. Následne sme vytvorili jednoduchý skript „ColorChange“, ako je znázornené v ukážke kódu 1. V skripte sme zmenili materiál štetca podľa farby gule. Tento skript sa spúšťal len pri vstupe triggeru s objektom. V parametroch každej gule sme priradili skriptu samotný štetec a odpovedajúcu farbu. Táto zmena avšak nastane až pri novej čiare a nezmení farbu čiary, ktorá sa aktuálne vytvára.

```
public class ColorChange : MonoBehaviour
{
    public GameObject brush;
    public Material material;

    private void OnTriggerEnter(Collider other)
    {
        brush.GetComponent<LineRenderer>().material = material;
    }
}
```

Úkážka kódu 1: Zmena farby



Obr. 6: Gule na zmenu farby písma

### 3.3.2 Úpravy písania - zmena nástroja

Vytvorili sme nový 3D objekt gule a umiestnili sme ho k ostatným modifikátorom písania, ako je znázornené na obrázku číslo 7 b). Priradili sme mu špeciálny materiál a aktivovali sme mu funkciu is Trigger. Vytvorili sme nový skript s názvom „WriteChange“. Tento skript slúži ako prepínač medzi módami písania a spúšťa sa len na začiatku kolízie. Obsahuje jednoduchý Boolean, ktorý používame na vyjadrenie stavu písania perom true alebo písania prstom false.

### 3.3.3 Úpravy písania - premazanie plátna

Tak ako v predošlom kroku sme vytvorili nový 3D objekt gule, ako je znázornené na obrázku číslo 7 c). Tento objekt bude slúžiť na premazávanie plátna, ak by používateľ potreboval plátno zresetovať a písať nanovo. Vytvorili sme nový skript s názvom „Clean“ a priradili ho objektu. Následne sme skript upravili tak, aby pri začiatku kolízie spočítal všetky vytvorené čiary. Robíme to pomocou sčítania všetkých detských objektov, ktoré vznikli pod prázdny objektom DrawT. Následne sa spustí druhá časť skriptu, ktorá pomocou for cyklu postupne vymaže všetky detské objekty. Mazanie objektov robíme pomocou príkazu Destroy().

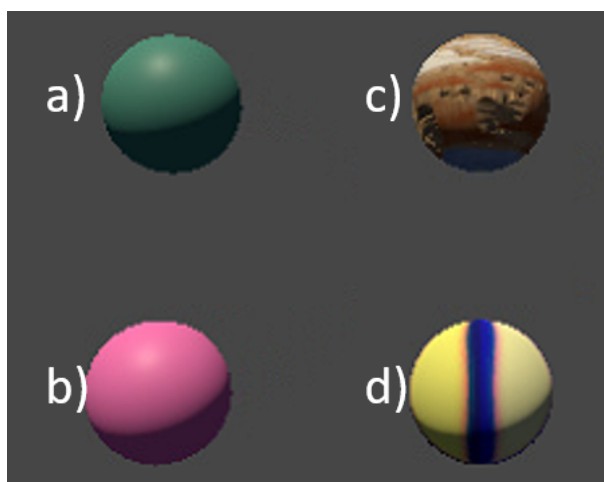
### 3.3.4 Úpravy písania - uloženie plátna

Na ukladanie plátna do operačného systému sme vytvorili dva objekty. Prvý objekt je 3D objekt gule, ktorý sme umiestnili k ostatným objektom slúžiacich ako spúšťače, ako je znázornené na obrázku číslo 7 d). Tento objekt slúži ako spúšťač funkcie uloženia plátna. Druhý objekt je objekt Camera, pomocou ktorej budeme zaznamenávať plátno. Kameru

nastavíme ako detský objekt plátna. Umiestníme kameru tesne nad plátno tak, aby bola rovnobežná s plátnom. Upravíme rozmery Kamery tak, aby odpovedali rozmerom plátna. Vytvoríme komponent skript na kamere s názvom CameraShot. Pri spustení tohto skriptu sa uloží čas operačného systému a vykreslí sa jeden snímok z objektu kamery. Snímok sa následne pretransformuje do formátu PNG obrázku a uloží sa v projekte s názvom času operačného systému.

### 3.3.5 Uprava plátna - škálovanie

Ďalšou interakciou v našej práci je škálovanie plátna pomocou gesta. Detekcia gesta sa spúšťa pomocou 3D objektu guľa, ktorý je umiestnený pri ostatných modifikátoroch úpravy písania, ako je znázornené na obrázku číslo 7 a). Viac informácií o využití gest ako funkcií bude popísané v podkapitole 3.4.4.



Obr. 7: Gule na pokročilé funkcie, a)škálovanie plátna, b)prepínanie písma, c)zmazanie plátna, d) ukladanie plátna

## 3.4 Gestá vo virtuálnom prostredí

Gestá sú nevyhnutnou súčasťou virtuálneho prostredia, používajú sa na interakciu so svetom. Za gestá môžeme považovať pohyby alebo polohy rúk, prstov, či hlavy. Po zaznamenaní špecifického gesta môžeme vykonať zvolenú akciu napríklad pohyb, špeciálna funkcia a rôzne interakcie s objektami ako je výber, posúvanie, chytanie, či modifikácie.

### 3.4.1 Vytvorenie a detegovanie gesta

V našom projekte sme si vytvorili tri vlastné nové gestá, ktoré sme následne využívali ako spúšťače skriptov. Na to, aby sme gesto mohli využívať, treba naučiť skript o aké gesto ide. Na to sme vytvorili nový prázdny objekt s názvom Detektor, ktorý bude slúžiť na účely

vytvárania a detekcie rôznych gest. Vytvorili sme v objekte Detektor nový komponent skript a upravili ho tak, aby mal dva módy, medzi ktorými vieme prepínať pomocou booleanu. V prvom móde skript učíme tri nové gestá a v druhom móde naučené gestá detegujeme.

Na štarte projektu skript má zadané dve informácie a to s akou presnosťou má detegovať gesto a ruku, ktorej gestá bude detegovať. Samotný štart skriptu je schválne oneskorený, pretože generovanie rúk v projekte nie je okamžité a interval prvotného generovania vie byť aj dve sekundy. Po oneskorení si v liste vypíšeme všetky vygenerované kosti ruky, ktorú sme na začiatku zadali.

Keď je projekt spustený, tak každý jeden snímok obnovy spúšťa druhú časť skriptu na detekciu. V každom snímku, v ktorom kamera vidí danú ruku, sa zaznamená kostra ruky. Dočasne sa vytvorí kópia kostry ruky. Následne sa postupne porovnávajú lokálne pozície skopírovanej ruky s jednotlivými gestami a vypočítavajú sa odchýlky. Ak daná odchýlka je nižšia ako presnosť, ktorú sme zadali na začiatku, tak si skript zapamätá novú odchýlku a porovnáva ju s ďalšími gestami. Na konci cyklu gesto s najnižšou odchýlkou prehlasujeme ako správne.

V prípade, ak chce používateľ uložiť gesto, ktoré práve znázorňuje, musíme prepnúť skript do učiaceho módu. V tomto móde prebiehajú rovnaké funkcie ako v detegovanom móde, ale navyše používateľ vie ukladať gestá stlačením medzerníka. Tým sa vytvorí v množine gest nové gesto s pozíciami kostí.

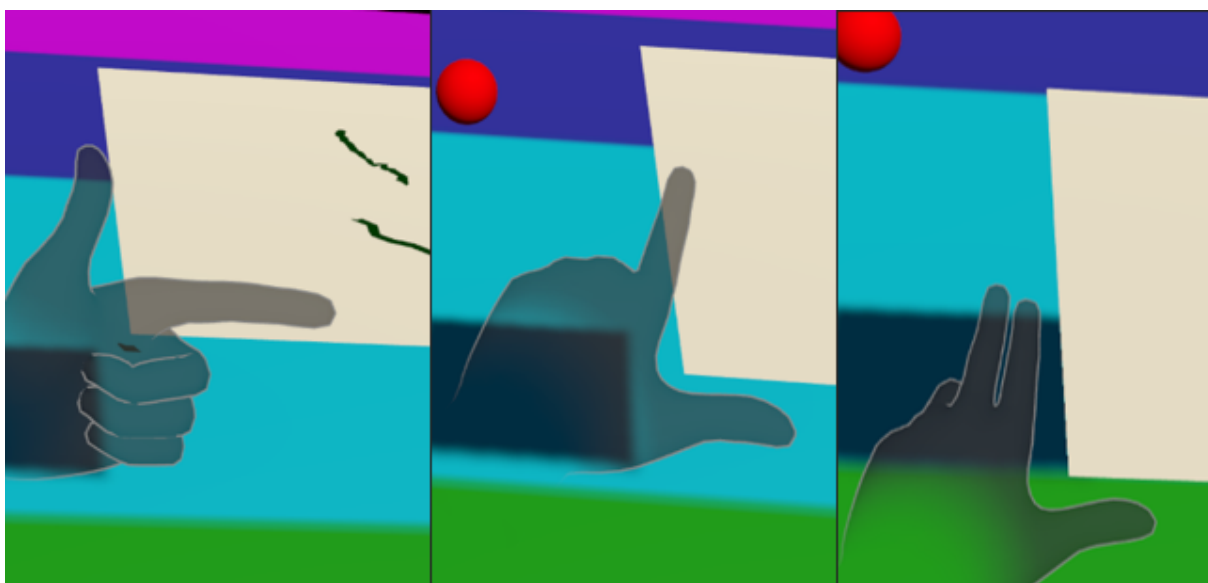
Po naučení skriptu vieme jednotlivé gestá analyzovať a pomenovať. Ak nám dáta vyhovujú a po naučení ich skript deteguje správne, tak môžeme jednotlivé gestá použiť ako spínače pre nové skripty a tým naplno využívať prostredie nášho VR projektu.

### **3.4.2 Úprava plátna pomocou gesta**

V nasledujúcej časti sme využili novovytvorené gestá ako spúšťače skriptov na rozšírenie interakcie používateľa s virtuálnym prostredím. Gestá sme využili dvoma spôsobami. Prvý spôsob je pohyb plátna po zvislej osi a tým sa nastaví úroveň plátna podľa výšky používateľa. Druhý spôsob je škálovanie plátna pomocou dvoch bodov, vďaka ktorých si vieme plátno ľubovoľne zväčšiť alebo zmenšiť podľa potreby spolu s už namaľovanými čiarami.

### **3.4.3 Pohyb plátna**

Ako prvú modifikáciu sme si zvolili úpravu výšky plátna. Pre vyššiu interakciu s používateľom sme si zvolili úpravu daného parametru pomocou gesta. Vytvorili sme si nový



Obr. 8: Znázornenie vytvorených gest

komponent skript v objekte plátna s názvom „Move“, ktorý sme znázornili v ukážke 2. Skript sme upravili tak, aby sa púšťal len v prípade ak, kamera zaregistruje jedno z vytvorených gest ľavej ruky. V prípade ak zaregistruje požadované gesto, tak vypočíta stredný bod plátna, pričom zvislú ypsilonovú súradnicu vynuluje. Následne pripočíta k aktuálnej pozícii koreňa ruky túto hodnotu spolu s pevne nastavenou odchýlkou (0,085 ; 0 ; 0,055). Výsledkom je posunutie plátna tak, aby roh plátna bol zároveň s rohom gesta ľavej ruky, pričom plátno ostalo vodorovné. Použite danej funkcie je dynamické, čiže plátnom vieme pohybovať plynulo bez nutného znovuznázornenia gesta pri každom pohybe.

```
public void MoveN()
{
    Vector3 halfPlane = (transform.localScale / 2) * 10;
    halfPlane.y = 0;
    transform.position = position.position + (offset + halfPlane);
}
```

Úkážka kódu 2: Pohyb plátna pomocou gesta

### 3.4.4 Škálovanie plátna

Ako druhú modifikáciu sme si zvolili dynamickú úpravu pozície plátna podľa potreby používateľa. Vytvorili sme nový prázdny objekt „MovePlatnoParent“ a v ňom sme vytvorili dva 3D objekty gulí. Tieto gule sme nazvali „LowerBall“ a „UpperBall“. Umiestnili sme



ich do dvoch oproti stojacich rohov plátna a upravili ich škály na 0,03 vo všetkých rozmeroch. Obom objektom gúl sme priradili dva skripty z Oculus Integration balíčka a to skripty „Grabbable(Script)“ a „Hand Grab Interactable(Script)“, kde sme nastavili Pointable Element a Rigidbody na objekt jednotlivej gule. Taktiež sme priradili komponent Rigidbody guliam. V Rigidbody sme guliam deaktivovali gravitáciu a povolili sme im kinematiku. Ako posledné sme týmto guliam zamrazili pozíciu Y, čím zaručíme rovnakú výšku gúl aj keď s nimi budeme manipulovať. Pomocou integrovaných skriptov používateľ vie s objektami interagovať, vložiť ich do rúk a posúvať ich.

V rodičovskom objekte MovePlatnoParent sme vytvorili nový komponent skript s názvom „PlatnoScale“, ktorý sme znázornili v ukážke 3. V skripte sme každý snímok načítali pozície gúl LowerBall a UpperBall. Vytvárame novú pozíciu newPosition, ktorá je v strede medzi súradnicami gúl a vytvárame novú škálu newScale podľa zmeny vzdialeností týchto gúl, pričom škálu v smere Y nechávame na hodnote 1. Následne nový rozmer a škálu aplikujeme na naše plátno. Tým plátno buď zväčšíme alebo zmenšíme v reálnom čase podľa potreby používateľa.

```
void Update()
{
    upperBall.position = new Vector3 (upperBall.position.x,
        platno.position.y, upperBall.position.z);
    lowerBall.position = new Vector3 (lowerBall.position.x,
        platno.position.y, lowerBall.position.z);
    Vector3 newPosition = (upperBall.position + lowerBall.position) / 2;
    Vector3 newScale = (upperBall.position - lowerBall.position) / 10;
    newScale.y = 1;
    platno.localScale = newScale;
    platno.position = newPosition;
}
```

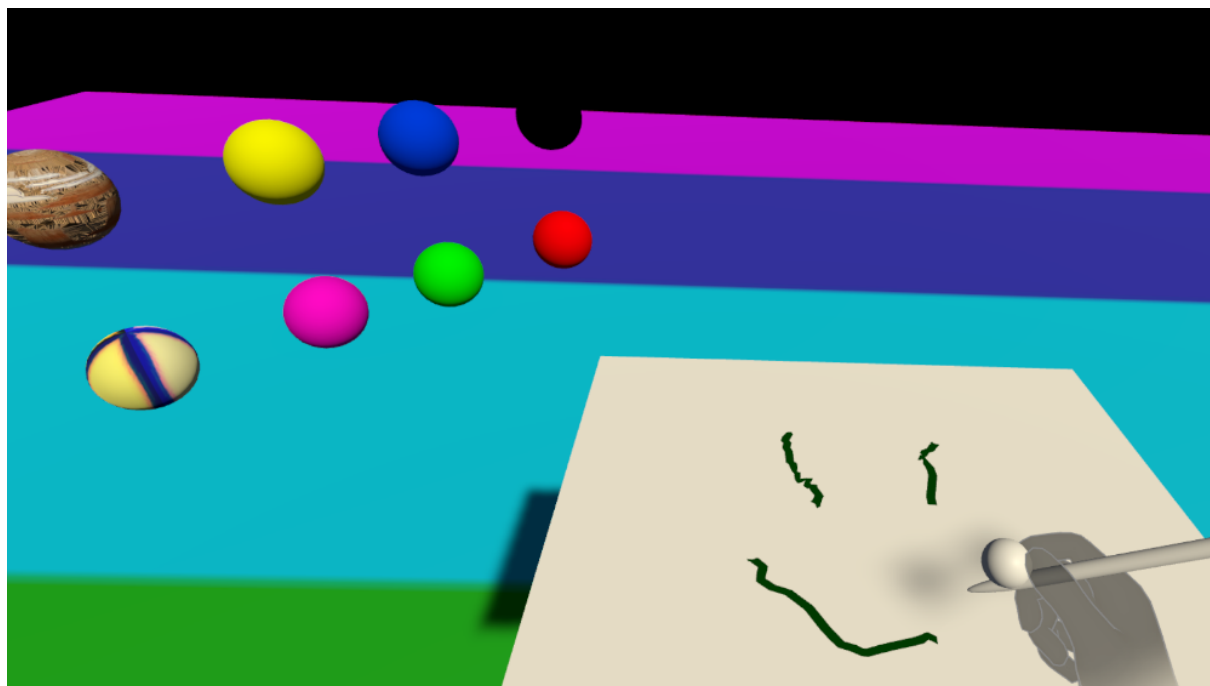
Úkážka kódu 3: Škálovanie plátna

## 4 Diskusia

Našou hlavnou úlohou bolo vytvorenie systému pre vyhodnocovanie písania pomocou grafického tabletu vo virtuálnom 3D prostredí. Považujeme úlohu za vyriešenú a táto práca poskytuje dôkaz funkčnosti konceptu.

Pri testovaní s dobroľníkmi sme odhalili viacero výhod a nevýhod nášho projektu. Testy trvali približne 15 minút. Po krátkej inštrukcii sa používatelia začali orientovať v prostredí. Používali základné, ale aj pokročilé funkcie nášho projektu. Písali rôznymi čiarami, premazávali a ukladali plátna, posúvali plátno podľa potrieb a aj striedavo písali perom a prstom. Názorná ukážka čo dobrovoľníci videli je v obr 9.

Na začiatku mal používateľ pocit prekvapenia z neznámeho prostredia. Používatelia, ktorí mali VR headset prvýkrát na sebe, pocítovali menšie nevoľnosti. Ostatní nemali problémy so zdravím a orientáciou. Písanie perom sa im javilo ako prirodzené, presnejšie odrážalo predstavu. Používateľské rozhranie im nepripadalo plne intuitívne, čiže sa museli pýtať a sústrediť sa na jeho ovládanie. Pri rozhodovaní sa, či je lepšie písanie perom alebo prstom, boli používatelia rozdelení na dva totožné tábory. Na konci testovania používatelia vyhodnotili náš projekt pozitívne a perspektívne na možné budúce praktické využitie.



Obr. 9: pohľad používateľa

Úspešne sme si naštudovali a analyzovali tému virtuálnej reality a jej okolie. Problémy nastali pri testovaní prepojenia grafického tabletu s VR zariadením. V aktuálnom stave Oculus Quest 2 nepovoľuje komunikáciu väčšine komponentov, ktoré nie sú určené priamo na VR. Pôvodne táto funkcia bola dostupná, ale bezpečnostnými aktualizáciami sa postupne kompatibilita znížila na skupinu vybraných klávesníc a myší. Je možné, že v budúcnosti sa táto funkcionality obnoví a bude možné využiť grafický tablet priamo s VR zariadením.

Ďalším väčším problémom bolo využitie prvkov augmentovanej reality. V našej práci sme sa pokúšali o vygenerovanie virtuálneho tabletu pomocou QR značiek umiestnených na tablete, ktoré by boli snímané buď kamerami na VR zariadení alebo externe pred používateľom. Tak ako Oculus Quest 2 obmedzil kompatibilitu zariadení na pripojenie, tak aj obmedzil použitie pokročilých funkcií predným kamerám. Pre použitie externých kamier by bolo potrebné využitie špeciálnych priestorových kamier na správnu aplikáciu triangulácie. Pre krátky zostávajúci čas práce sme si zvolili možnosť vynechania AR prvkov a venovali sme sa čisto virtuálnej realite.

## 5 Záver

Analýzovali sme vybrané prvky virtuálnej reality. V prvom rade sme analyzovali jej využitie v rôznych odvetviach ako sú vizualizácie, simulácie, herný priemysel, virtuálne pracovne, či virtuálne učebne.

Ako druhé sme analyzovali vybrané spôsoby interakcie vo virtuálnej realite. Zamerali sme sa na zaznamenávanie a rozpoznávanie gest. Zisťovali sme možnosti samotného zaznamenávania gest pomocou viacerých metód, ako je LeapMotion či DataGlove. Taktiež sme analyzovali ďalšie možnosti vstupov a interakcií vo virtuálnom prostredí ako sú klávesnice, tablety, grafické tablety a špeciálne hardvéry ako sú perá.

Ďalej sme preverili vybrané dostupné prostredia na vývoj 3D virtuálneho sveta. Zamerali sme sa na Unreal Engine, CryEngine a Unity Engine. Jednotlivé prostredia sme si preštudovali, zistili ich kompatibilitu s tvorbou virtuálneho sveta. Vyhodnotili sme ich výhody a nevýhody. Pre značné výhody a hlbšiu predošlú znalosť sme si vybrali Unity Engine.

Pred samotnou tvorbou projektu a testovaním funkcií sme vytvorili návrh riešenia našej práce s možnosťami prepojenia grafického tabletu s virtuálnou realitou.

Nainštalovali sme prostredie Unity a potrebné rozšírenia na prácu pre 3D virtuálne prostredie. Taktiež sme nainštalovali potrebné ovládače a dodatočný softvér pre správne pracovanie počítača s VR zariadením a grafickým tabletom. Pre VR zariadenie sme museli vytvoriť účet a nakonfigurovať účet.

Navrhli sme možné prepojenia grafického tabletu s VR zariadením. Vytvorili sme testovacie projekty v Unity na určenie funkčnosti. Pomocou výstupov z testovacích projektov sme určili, že zapojenie pomocou počítača bolo vhodné pre náš projekt.

Vytvorili sme projekt, v ktorom sme vytvorili 3D virtuálnu miestnosť. V miestnosti sme implementovali a testovali funkcie. Zberali sme dáta z grafického tabletu, hlavy a rúk používateľa. Skúšali sme viacero metód písania čiar, z ktorých nám najviac vyhovoval komponent Line Renderer. Vytvorili sme metódu písania na virtuálne plátno pomocou dát z grafického tabletu. Následne sme pridávali funkcionality ako písanie pomocou prsta, zmena farby písma, mazanie a ukladanie plátna. Taktiež sme vytvorili vlastné gestá, ktoré slúžili ako spúšťače skriptov na úpravu plátna.

V budúcej práci sa budeme venovať kompatibilitu prostredia voči vypracovanej aplikácii pre systémy Windows a Mac. Vybudovaní funkčnej aplikácie pre dané operačné systémy. V prípade sprístupnenia zo strany vydavateľa VR zariadenia, by sme celý projekt prekonvertovali do standalone režimu, čo znamená, že by sme nemuseli využiť počítač a

tablet by komunikoval s VR zariadením napriamo.

Taktiež sa budeme venovať rozšíreniu funkcií v projekte, ako je funkcia dynamickej šírky písma, či možnosť použitia viacerých pláten. Upravili by sme používateľské rozhranie v prostredí, aby bolo intuitívnejšie a jednoduchšie na použitie. Nakoniec by sme sa pokúsili rozšíriť myšlienku aplikovania augmentovanej reality a vytvorili tým zmiešanú realitu.

# Zoznam použitej literatúry

1. RIBARSKY, W., BOLTER, J., OP DEN BOSCH, A. a TEYLINGEN, R. van. Visualization and analysis using virtual reality. *IEEE Computer Graphics and Applications*. 1994, roč. 14, č. 1, s. 10–12. Dostupné z DOI: 10.1109/38.250911.
2. GUPTA, Sarthak, BAGGA, Siddhant a SHARMA, Deepak Kumar. Hand Gesture Recognition for Human Computer Interaction and Its Applications in Virtual Reality. In: *Advanced Computational Intelligence Techniques for Virtual Reality in Healthcare*. Ed. GUPTA, Deepak, HASSANIEN, Aboul Ella a KHANNA, Ashish. Cham: Springer International Publishing, 2020, s. 85–105. ISBN 978-3-030-35252-3. Dostupné z DOI: 10.1007/978-3-030-35252-3\_5.
3. PINTER, Csaba, LASSO, Andras, CHOUEIB, Saleh, ASSELIN, Mark, FILLION-ROBIN, Jean-Christophe, VIMORT, Jean-Baptiste, MARTIN, Ken, JOLLEY, Matthew A. a FICHTINGER, Gabor. SlicerVR for Medical Intervention Training and Planning in Immersive Virtual Reality. *IEEE Transactions on Medical Robotics and Bionics*. 2020, roč. 2, č. 2, s. 108–117. Dostupné z DOI: 10.1109/TMRB.2020.2983199.
4. JANG, Yeonju a PARK, Eunil. An adoption model for virtual reality games: The roles of presence and enjoyment. *Telematics and Informatics*. 2019, roč. 42, s. 101239. ISSN 0736-5853. Dostupné z DOI: <https://doi.org/10.1016/j.tele.2019.101239>.
5. FEREYDOONI, Nadia a WALKER, Bruce N. *Virtual Reality as a Remote Workspace Platform: Opportunities and Challenges*. 2020. Dostupné tiež z: <https://www.microsoft.com/en-us/research/publication/virtual-reality-as-a-remote-workspace-platform-opportunities-and-challenges/>.
6. RAJA, M a LAKSHMI PRIYA, GG. An Analysis of Virtual Reality Usage through a Descriptive Research Analysis on School Students' Experiences: A Study from India. *International Journal of Early Childhood Special Education*. 2021, roč. 13, č. 2.
7. NAKAI, Kohga, TERADA, Satoshi, TAKAHARA, Ayaka, HAGE, Dany, TUBBS, R Shane a IWANAGA, Joe. Anatomy education for medical students in a virtual reality workspace: A pilot study. *Clinical Anatomy*. 2022, roč. 35, č. 1, s. 40–44.
8. RAUTARAY, Siddharth S. a AGRAWAL, Anupam. Interaction with virtual game through hand gesture recognition. In: *2011 International Conference on Multimedia, Signal Processing and Communication Technologies*. 2011, s. 244–247. Dostupné z DOI: 10.1109/MSPCT.2011.6150485.

9. KHUNDAM, Chaowanan, VORACHART, Varunyu, PREEYAWONGSAKUL, Patibut, HOSAP, Witthaya a NOËL, Frédéric. A Comparative Study of Interaction Time and Usability of Using Controllers and Hand Tracking in Virtual Reality Training. *Informatics*. 2021, roč. 8, č. 3. ISSN 2227-9709. Dostupné z DOI: 10.3390/informatics8030060.
10. SCHÄFER, Alexander, REIS, Gerd a STRICKER, Didier. Controlling Teleportation-Based Locomotion in Virtual Reality with Hand Gestures: A Comparative Evaluation of Two-Handed and One-Handed Techniques. *Electronics*. 2021, roč. 10, č. 6. ISSN 2079-9292. Dostupné z DOI: 10.3390/electronics10060715.
11. KHUNDAM, Chaowanan. First person movement control with palm normal and hand gesture interaction in virtual reality. In: *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. 2015, s. 325–330. Dostupné z DOI: 10.1109/JCSSE.2015.7219818.
12. TSAI, Chia-Chun, KUO, Chih-Chia a CHEN, Yen-Lun. 3D Hand Gesture Recognition for Drone Control in Unity. In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. 2020, s. 985–988. Dostupné z DOI: 10.1109/CASE48305.2020.9216807.
13. XU, Deyou. A Neural Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG. In: *18th International Conference on Pattern Recognition (ICPR'06)*. 2006, zv. 3, s. 519–522. Dostupné z DOI: 10.1109/ICPR.2006.109.
14. LI, Yang, HUANG, Jin, TIAN, Feng, WANG, Hong-An a DAI, Guo-Zhong. Gesture interaction in virtual reality. *Virtual Reality Intelligent Hardware*. 2019, roč. 1, č. 1, s. 84–112. ISSN 2096-5796. Dostupné z DOI: <https://doi.org/10.3724/SP.J.2096-5796.2018.0006>.
15. KANG, Taeseok, CHAE, Minsu, SEO, Eunbin, KIM, Mingyu a KIM, Jinmo. DeepHandsVR: Hand Interface Using Deep Learning in Immersive Virtual Reality. *Electronics*. 2020, roč. 9, č. 11. ISSN 2079-9292. Dostupné z DOI: 10.3390/electronics9111863.
16. *Tracked keyboards for Meta Quest Pro and Meta Quest 2* [online]. [cit. 2023-04-23]. Dostupné z : <https://www.meta.com/help/quest/articles/headsets-and-accessories/meta-quest-accessories/tracked-keyboards-meta-quest/>.
17. *Wacom VR Pen* [online]. [cit. 2023-04-23]. Dostupné z : <https://developer.wacom.com/en-us/wacomvrpen>.

18. DREY, Tobias, GUGENHEIMER, Jan, KARLBAUER, Julian, MILO, Maximilian a RUKZIO, Enrico. VRSketchIn: Exploring the Design Space of Pen and Tablet Interaction for 3D Sketching in Virtual Reality. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu, HI, USA: Association for Computing Machinery, 2020, s. 1–14. CHI '20. ISBN 9781450367080. Dostupné z DOI: 10.1145/3313831.3376628.
19. CIEKANOWSKA, Agata, KISZCZAK - GLIŃSKI, Adam a DZIEDZIC, Krzysztof. Comparative analysis of Unity and Unreal Engine efficiency in creating virtual exhibitions of 3D scanned models. *Journal of Computer Sciences Institute*. 2021, roč. 20, s. 247–253. Dostupné z DOI: 10.35784/jcsi.2698.

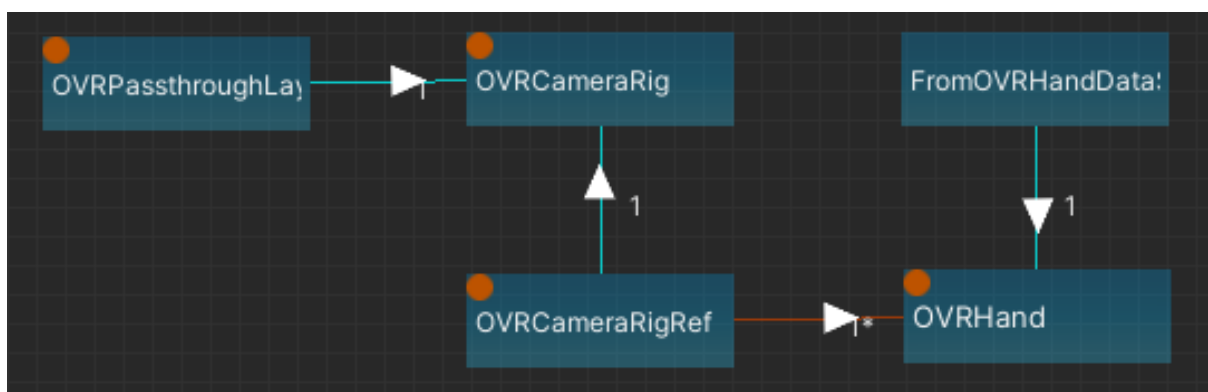


# Prílohy

A	Dokumentácia . . . . .	II
B	Zdrojový kód . . . . .	IV

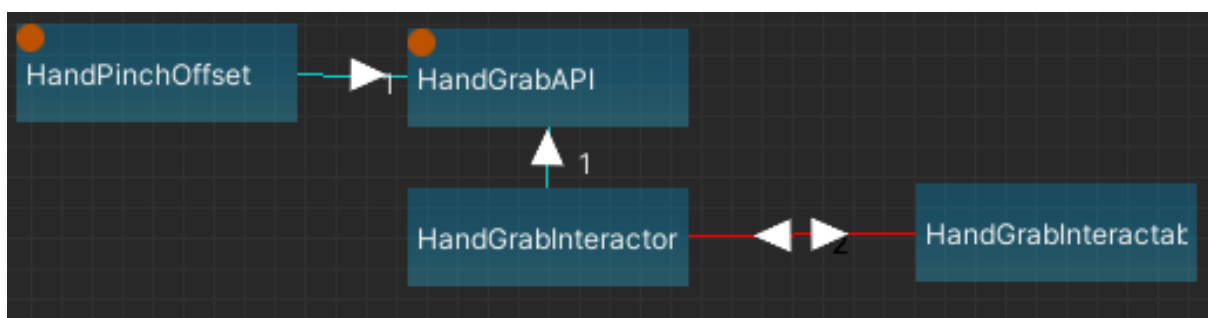
## A Dokumentácia

Na obrázku A.1 je znázornený diagram kamery. Slúži na samotné generovanie pohľadu používateľa a detegovanie rúk v zornom poli. Pomocou funkcie `OVRPassthroughLayer` sme testovali prepnutie pohľadu z virtuálnej reality na realitu zmiešanú, čiže by používateľ videl reálne prostredie okolo seba.



Obr. A.1: pohľad používateľa

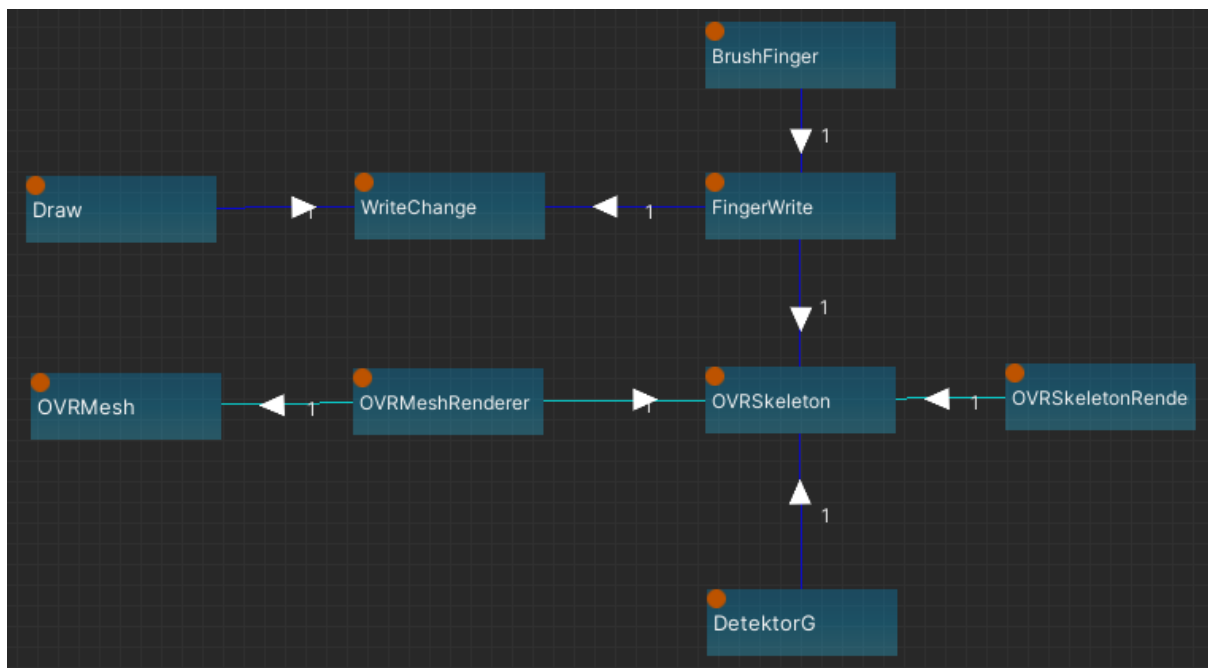
Na chytanie objektov nám slúžil diagram znázornený na obrázku A.2. Využili sme základnú funkcionálnosť Oculus rozšírenia v Unity. Pomocou základných gest vieme uchopiť predmety, na ktorých je funkcia `HandGrabInteractable`.



Obr. A.2: chytanie objektov

Generovanie rúk a písanie je znázornené na obrázku A.3. Hlavný uzol je `OVRSkeleton`, ktorý prijíma dáta na generovanie kostry rúk používateľa a aj samotný povrch rúk. `DetektorG` vytvára gestá podľa kostry `OVRSkeleton` a zároveň deteguje kosť ruky, či neboli vykonané gestá. `FingerWrite` zabezpečuje písanie prstom na plátne na základe pozícií kosti prsta `OVRSkeleton`. `Draw` funkcia zabezpečuje písanie perom podľa pozícií pera na grafickom tablete. Funkcia `WriteChange` slúži na prepínanie `Draw` funkcie, čím

vieme vypnúť alebo zapnúť písanie perom. Na prepínanie hodnôt WriteChange používame spúšťač BrushFinger, ktorý je umeistený na objekte.



Obr. A.3: ruky - písanie, gestá a generovanie

Na ukladanie plátna sme využili jednoduchý diagram z obrázku A.4. ScreenShot je spínač, pracujúci na báze spúšťača pri kolízií objektov, ktorý spúšťa skript CameraShot. CameraShot je umiestnený na ortogonálnej kamere, ktorá je umiestnená nad plátnom. Pri každom spustení skript vytvorí obrázok, ktorý následne uloží v operačnom systéme.



Obr. A.4: ukladanie plátna

## B Zdrojový kód

Našu prácu sme uverejnili na portál GitHub. Nachádzajú sa tam tri položky. Prvý z nich je adresár .gitignore pre prostredie Unity, ktorý upravuje ukladanie Unity prostredia v Gite. Druhý adresár je naša uložená diplomová práca slúžiaca ako dokumentácia vytvoreného projektu. Nakoniec samotný Unity projekt, ktorý sme vypracovali.

Zdroj na Git Hub:

<https://github.com/SkorecFilip/Diplomova-praca>