



**ČESKÉ VYSOKÉ  
UČENÍ TECHNICKÉ  
V PRAZE**

**F3**

**Fakulta elektrotechnická  
Katedra počítačové grafiky a interakce**

**Bakalářská práce**

# **2D kreslení na papír ve virtuální realitě**

**Matyáš Koval'**

**Obor: Počítačové hry a grafika**

**Květen 2022**

**Vedoucí práce: Ing. David Sedláček, Ph.D.**



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Koval'** Jméno: **Matyáš** Osobní číslo: **483737**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**  
Studijní program: **Otevřená informatika**  
Specializace: **Počítačové hry a grafika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**2D kreslení na papír ve virtuální realitě**

Název bakalářské práce anglicky:

**2D painting in VR**

Pokyny pro vypracování:

Prostudujte možnosti realizace kreslení na papír ve virtuální realitě, a seznamte se Langweilovým modelem Prahy a postupy, jak byl vytvořen [7].

Navrhněte interakční techniky připomínající tvorbu domů Langweilova modelu (kreslení na papír tužkou, perem, štětcem s barevnou tuší, ořezání papíru a lepení částí modelu). Techniky navrhněte jak volné (free hand), tak omezené pomocí vodítek (guide lines, constraints). Uživatel může kreslit na zvětšený papír oproti reálu nebo v měřítku. Předpokládejte, že uživatel bude sedět u reálného stolu, který mu bude vytvářet pasivní haptickou zpětnou vazbu ve virtuálním světě (stůl ve VR uvidím na stejném místě).

Navržené techniky implementujte v herním enginu Unity a otestujte s vybranou skupinou uživatelů (po dohodě s vedoucím práce).

Implementované techniky zakomponujte do již existujícího zážitku nazvaného "Tvorba Langweilova modelu Prahy" [5] a přizpůsobte je vzhledem tomuto virtuálnímu prostředí. Následně změňte výkonu VR scény (např. pokles snímkové frekvence, zpoždění a prodlevu systému, paměťové nároky) a případně realizujte kroky pro dosažení plynulé (realtime) odezvy.

Seznam doporučené literatury:

- 1] Jason Jerald. 2015. The VR Book: Human-Centered Design for Virtual Reality. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA.
- 2] Joseph J. LaViola, Jr. et al. 3D User Interfaces: Theory and Practice, second edition. 2017. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
- 3] Steven M. LaValle - Virtual Reality, Cambridge University Press 2016
- 4] J. Žára, B. Beneš, J. Sochor, P. Felkel, Moderní počítačová grafika, 2. vydání. Computer Press, 2004.
- 5] Lukáš Pospíšil, Virtuální zážitek „Tvorba Langweilova modelu Prahy“. BP ČVUT FEL 2021.
- 6] Filip Suchý, Sledování reálných objektů ve virtuální realitě. DP ČVUT FEL 2021.
- 7] Kateřina Bečková, Miroslav Fokt. Svědectví Langweilova modelu Prahy. Schola Ludus Pragensia 1996.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. David Sedláček, Ph.D. katedra počítačové grafiky a interakce FEL**

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **08.02.2022** Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. David Sedláček, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## Poděkování / Prohlášení

Chtěl bych poděkovat vedoucímu své práce Ing. David Sedláčkovi Ph.D. za vedení, trpělivost, pravidelné konzultace, ochotu a cenné rady v průběhu této práce. Dále bych chtěl poděkovat své rodině, přátelům a v neposlední řadě i spolužákům, ze kterých se v průběhu studia přátelé stali, za jejich podporu.

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu a zdroje, a to v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Kladně, 19. května 2022

.....

## Abstrakt / Abstract

Cílem této bakalářské práce je navrhnout a prototypovat interakční techniky, které by připomínaly tvorbu domů Langweilova modelu Prahy a následně je zakomponovat do již existujícího virtuálního zážitku jménem „Tvorba Langweilova modelu Prahy“.

Práce je rozdělena do kapitol. V Úvodu je výčet požadavků práce a několik základních pojmů, potřebných v kontextu této práce. Následuje kapitola Průzkum, kde dojde k prozkoumání možností implementace požadavků. V kapitole Návrh, dojde k analýze těchto možností a návrhu jejich implementace, která je popsána v kapitole Implementace. Kapitola Testování popisuje procesy testování implementace, které byly využity a jejich výsledky. Finální kapitolou je Závěr, kde jsou sepsány závěry vyplývající z testování a ohodnocení práce autorem.

Výsledkem práce je ucelený virtuální zážitek, který uživateli poskytne možnost vyzkoušet si ve virtuální realitě za pomoci výše zmíněných interakčních technik tvorbu kusu Langweilova modelu Prahy v ateliéru na půdě Klementina.

**Klíčová slova:** Virtuální realita, Unity 3D, interakční techniky, kreslení ve VR, Langweilův model Prahy

The goal of this bachelor's work is the design and prototyping of interaction techniques, that would closely resemble the creation of houses for the Langweil's model of Prague and their implementation into an already existing virtual experience called 'VR experience: „Langweil Model of Prague Creation“.

The work is divided into chapters. The Introduction contains a list of the requirements for this work and a few basic terms, needed in context of this work. Following is the Research chapter, which is dedicated to the research of options for the implementation of the requirements. In the Design chapter, I will analyse these options and design their implementation, which is described in the Implementation chapter. The Testing chapter describes the processes of testing the implementation and their results. The Final chapter is the Conclusion, which contains the conclusions made from the testing results and the evaluation of the work by the author.

The result of this work is a complete virtual experience, that allows the user to try and utilise the above mentioned techniques to create a part of Langweil's model of Prague in virtual reality in an atelier in the attic of Klementinum.

**Keywords:** Virtual reality, Unity 3D, interaction techniques, painting in VR, Langweil's model of Prague

**Title translation:** 2D painting in VR

# Obsah /

<b>1 Úvod</b> .....	1
1.1 Virtuální realita .....	1
1.2 Unity .....	1
1.3 Langweilův model Prahy .....	2
1.4 Požadavky .....	2
1.4.1 Funkční požadavky .....	2
1.4.2 Nefunkční požadavky .....	3
<b>2 Průzkum možností implementace</b> .....	4
2.1 Kreslení .....	4
2.1.1 Unity metoda SetPixel .....	4
2.1.2 Využití shaderů .....	4
2.1.3 Využití veřejně dostupných metod .....	5
2.2 Řezání papíru .....	5
<b>3 Návrh implementace</b> .....	6
3.1 Analýza projektu DokoDemoPainter .....	6
3.1.1 Návrh štětce .....	7
3.1.2 Návrh škálování .....	7
3.1.3 Návrh kreslení s vodítky .....	7
3.2 Analýza projektu Splatoon's Ink System— Mix and Jam .....	8
3.2.1 Návrh štětce .....	8
3.2.2 Návrh škálování .....	9
3.2.3 Návrh kreslení s vodítky .....	9
3.3 Ezy-Slice .....	9
3.4 Návrh zážitku .....	10
<b>4 Implementace</b> .....	11
4.1 Implementace DokoDemoPainter .....	11
4.1.1 Implementace štěce .....	11
4.1.2 Virtuální realita .....	11
4.1.3 Kreslení s vodítky .....	12
4.1.4 Možnost škálování textury .....	12
4.2 Implementace Splatoon's Ink System— Mix and Jam .....	13
4.2.1 Implementace štěce .....	13
4.2.2 Virtuální realita .....	13
4.2.3 Kreslení s vodítky .....	14
4.2.4 Možnost škálování textury .....	14
4.3 Implementace do uceleného zážitku „Tvorba Langweilova modelu Prahy“ .....	14
4.3.1 Scéna .....	14
4.3.2 Kreslení .....	15
4.3.3 Mazání .....	17
4.3.4 Řezání .....	17
4.3.5 Škálování papíru .....	19
4.3.6 Vkládání do modelu .....	19
<b>5 Testování</b> .....	21
5.1 Proces testování zážitku .....	21
5.1.1 Výsledky z dotazníků v grafech .....	21
5.1.2 Výsledky uživatelského testování .....	25
5.2 Výkonnostní testování .....	25
5.2.1 Testování projektu „Tvorba Langweilova modelu Prahy“ .....	25
5.2.2 Testování aktuálního zážitku .....	26
5.2.3 Výsledky výkonnostního testování .....	26
<b>6 Závěr</b> .....	27
6.1 Požadavky implementace .....	27
6.2 Závěry z uživatelského testování .....	27
6.3 Zhodnocení implementace autorem .....	28
<b>Zdroje</b> .....	29
<b>A Dotazník vyplněný testovacími uživateli</b> .....	31
<b>B Readme soubor</b> .....	37

## Tabulky / Obrázky

<b>5.1.</b> Výsledky výkonnostního testování projektu „Tvorba Langweilova modelu Prahy“ ...	25
<b>5.2.</b> Výsledky výkonnostního testování projektu „2D kreslení na papír ve virtuální realitě“ ...	26
<b>2.1.</b> Screenshot Shadergraphu .....	5
<b>3.1.</b> Screenshot Vrstev barev .....	7
<b>3.2.</b> Screenshot Segmentace čar .....	9
<b>3.3.</b> atelierScreenshot .....	10
<b>4.1.</b> Kreslení s Vodítky.....	12
<b>4.2.</b> Porovnání barev - Splatoon....	13
<b>4.3.</b> Porovnání osvětlení .....	15
<b>4.4.</b> Porovnání barev - Splatoon....	20
<b>5.1.</b> Graf odpovědí - Kterou rukou píšeš?.....	21
<b>5.2.</b> Graf odpovědí - Jak často používáš headset? .....	22
<b>5.3.</b> Graf odpovědí - Způsob ovládání headsetu .....	22
<b>5.4.</b> Graf odpovědí - Fyzické potíže po používání VR.....	22
<b>5.5.</b> Graf odpovědí - Vztah ke kreslení .....	23
<b>5.6.</b> Graf odpovědí - Lepení modelů .....	23
<b>5.7.</b> Graf odpovědí - Tvorba grafického obsahu .....	23
<b>5.8.</b> Graf shrnující výsledky otázek se škálou.....	24
<b>6.1.</b> Screenshot z uživatelského testování .....	28



# Kapitola 1

## Úvod

Cílem této práce je prozkoumání možností a následná implementace interakčních technik pro kreslení na papír (a další potřebné interakce) ve virtuální realitě a jejich následné využití k vytvoření autentického zážitku tvorby části Langweilova modelu Prahy z 19. století.

Implementace těchto a dalších interakčních technik sloužících k poskytnutí co nejautentičtějšího zážitku z tvorby části modelu bude zasazena do již existujícího zážitku „Tvorba Langweilova modelu Prahy“ [1], který byl vytvořen v rámci bakalářské práce na ČVUT, Lukáše Pospíšila.

Motivací tvorby tohoto zážitku je poskytnout běžnému člověku možnost vyzkoušet si náročnost tvorby části Langweilova modelu tehdy dostupnými prostředky a následnému porovnání vlastní práce ve virtuální realitě s kusem naskenovaného modelu, do kterého následně uživatel vloží část modelu, kterou vytvořil on.

### 1.1 Virtuální realita

*„Virtuální realita je definována jako počítačem vytvořené prostředí, které může uživatel zažít a interagovat s ním, jako by bylo skutečné.“ [2]*

Tato citace dle mého názoru asi nejlépe vystihuje, co se schovává pod pojmem „virtuální realita“ v kontextu této práce. Bude se jednat o virtuální prostředí, v drtivě většině případů konkrétně o prostředí vytvořené v rámci práce „Tvorba Langweilova modelu Prahy“, se kterým uživatel může interagovat za pomoci virtuálně-realistických ovladačů. Cílem této práce je, aby interakce, které uživatel provádí co nejvíce připomínaly interakce v reálném světě a uživatel se co nejlépe mohl vžít do virtuálního zážitku.

Virtuální realita je kontextu této práce také spojena se zařízeními, kterými se uživateli virtuálně-realistní prostředí zprostředkovává a kterými s ní může interagovat. Těmito zařízeními jsou náhlavní displeje (head-mounted display) a virtuálně-realistní ovladače (controller). Během vývoje této práce bylo využito dvou různých headsetů a to headsetu Acer Windows Mixed Reality AH101 a headsetu Oculus Quest 1.

### 1.2 Unity

Unity, nebo přesněji Unity 3D je mezi-platformový (cross-platform) herní engine umožňující uživateli vytvářet hry či herní zážitky v 2D a 3D a poskytuje aplikační rozhraní (API) pro vytváření skriptů v programovacím jazyce C#.

Poskytuje také několik různých vykreslovacích pipeline (render pipeline), mezi kterými je i Universal Render Pipeline (také URP - více o této pipeline je zmíněno v 4.2.2), které je v rámci tohoto projektu využíváno.

Vykreslovací pipeline je model popisující, jakým způsobem se 3D herní objekty vykreslí na 2D obrazovku, ať už virtuálně-realistního headsetu nebo počítače.

## 1.3 Langweilův model Prahy

„Langweilův model Prahy jako prostorové zobrazení města je ojedinělým uměleckým dílem a světovým unikátem svého druhu. Model z papírové lepenky na dřevěné konstrukci vytvořil v letech 1826-1837 sluha Univerzitní knihovny v Klementinu Antonín Langweil. Model obsahuje přes dva tisíce budov historického jádra Prahy v dokonalém realistickém provedení se všemi stavebními a zdobnými detaily fasád.“ [3]

Tento model byl v letech 2006-2008 zdigitalizován a jeho části jsou v digitální podobě využívány v ateliéru projektu „Tvorba Langweilova modelu Prahy“.

## 1.4 Požadavky

Součástí bakalářské práce je studie a realizace kreslení na papír ve virtuální realitě, návrh interakčních technik potřebných pro tvorbu domů Langweilova modelu (jak volných, tak omezených za pomoci vodítek) jako je například kreslení na papír perem, štětcem, možnost ořezávat papír a následného vložení do modelu k porovnání, které budou dávat dohromady ucelený virtuálně-reálný zážitek tvorby kusu Langweilova modelu Prahy.

Tvorba tohoto virtuálního zážitku bude probíhat za využití vývojové platformy Unity[4]. Tedy průzkum, návrh implementace i prototypování již probíhaly v rámci této platformy a jejích možností.

### 1.4.1 Funkční požadavky

#### ■ FP1 Kreslení na papír inkoustovým perem

Uživatel bude moci kreslit na papír inkoustovým perem.

#### ■ FP2 Kreslení na papír štětcem

Uživatel bude moci kreslit na papír štětcem.

#### ■ FP3 Kreslení na papír tužkou

Uživatel bude moci kreslit na papír tužkou.

#### ■ FP4 Možnost změny barvy štětce

Uživatel bude mít možnost měnit barvu, kterou štětcem kreslí.

#### ■ FP5 Ořezávání papíru

Uživatel bude moci ořezat papír, na který kreslí.

#### ■ FP6 Možnost znovu slepit rozřezaný papír

Uživatel bude moci rozřezaný papír následně opět slepit.

#### ■ FP7 Interakční techniky volné

Uživatel nebude nijak omezen během využívání těchto interakčních technik.

#### ■ FP8 Interakční techniky omezené

Při využívání těchto interakčních technik bude uživatel nějakým způsobem omezen. Například bude moci kreslit pouze rovné čáry.

#### ■ FP9 Vkládání papíru do modelu domu

Uživatel bude moci vložit papír, který má ve scéně k dispozici do modelu domu.

#### ■ FP10 Možnost měnit velikost papíru

Uživatel si bude moci zvětšit či zmenšit papír, na který kreslí.

#### ■ FP11 Mazání kreseb

Veškeré kresby bude uživatel moci v případě potřeby smazat.

#### ■ FP12 Zakomponování do existujícího zážitku „Tvorba Langweilova modelu Prahy“

Veškeré interakční techniky je třeba zakomponovat do existujícího zážitku „Tvorba Langweilova modelu Prahy“ a přizpůsobit je vzhledem tomuto virtuálnímu prostředí.

■ **FP13 Změřit výkon po zakomponování do existujícího zážitku**

Po zakomponování do existujícího zážitku je třeba změřit změnu výkonu virtuálně-reálné scény a případně realizovat kroky pro dosažení plynulé odezvy.

■ **1.4.2 Nefunkční požadavky**

■ **NFP1 Celý virtuálně-reálný zážitek bude implementován v herním enginu Unity**

Součástí požadavků je implementace všech požadavků v rámci herního enginu Unity. Tato návaznost je hlavně z důvodu, že již existující zážitek „Tvorba Langweilova modelu Prahy“ je implementována právě v Unity.

■ **NFP2 Virtuálně-reálný zážitek by měl být plynulý**

V případě, že při realizaci **FP13** bude zjištěn pokles výkonu, který by měl za příčinu menší než plynulou odezvu, je třeba výkon zážitku optimalizovat.

■ **NFP3 Virtuálně-reálný zážitek by měl být v rámci možností co nejautentičtější**

Tento požadavek je víceméně všeříkající. Žádná z implementovaných interakčních technik by neměla narušovat dojem věrohodnosti a realističnosti zážitku.

# Kapitola 2

## Průzkum možností implementace

### 2.1 Kreslení

Tento průzkum byl po konzultaci s vedoucím práce rozdělen na tři části. Implementace a následné vyzkoušení metody „SetPixel“ [5] dostupné v rámci platformy Unity, možnost využití shaderů ke kreslení do textur a průzkum veřejně dostupných metod, které jsou vytvořeny v, nebo umožňují kreslení do textur v Unity.

Texturou je v rámci této práce myšlena vizuální textura, tedy soubor (většinou v podobě obrázku), který určuje vizuální charakteristiky objektu, například jeho barvu, schopnost odrážet či pohlcovat světlo atp.

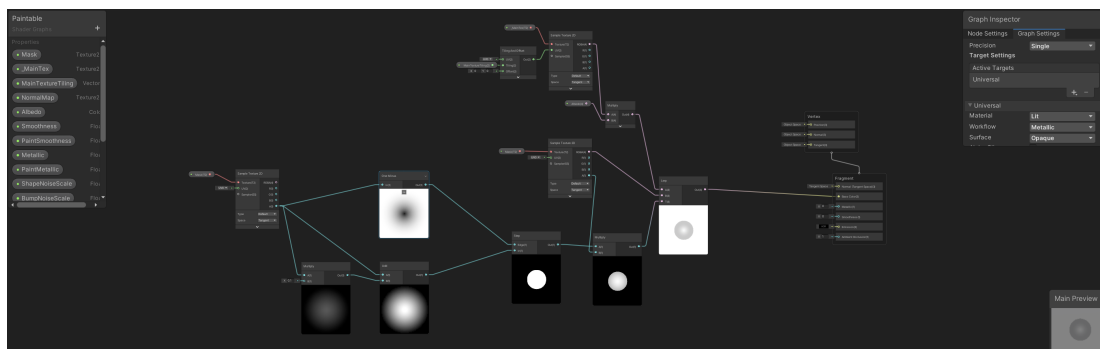
#### 2.1.1 Unity metoda SetPixel

Metoda „SetPixel“ je jednou z funkcí dostupných k úpravě 2D textur přímo v Unity. Při jejím volání je třeba uvést souřadnice X a Y pixelu v textuře, do kterého chceme zapisovat a barvu, na kterou chceme daný pixel přebarvit. Následně je nutné veškeré změny v texturách aplikovat, což je procesně nejnáročnější částí využití celé metody v závislosti na množství přepsaných dat.

Tato metoda je nejzákladnějším způsobem, jak implementovat kreslení do textur v Unity a jelikož se jedná o úpravu textur doslova pixel po pixelu. Ve chvíli, kdy bylo třeba využít tuto metodu k realizaci výše zmíněných požadavků, se ukázala být implementačně příliš složitou - bylo by třeba vytvořit celý framework (soubor tříd, funkcí a metod, které by dohromady poskytovaly funkcionalitu v tomto případě kreslení), který by využíval tuto metodu tak, aby výsledek alespoň připomínal kresbu štětcem či perem. Vzhledem k tomu, že samotné kreslení na papír bylo pouze jedním z požadavků práce, byl tento postup zavrhnut již ve fázi průzkumu.

#### 2.1.2 Využití shaderů

Shader [6] jako takový je program běžící na grafické kartě, který zpracovává všechny pixely a určuje, s jakými vlastnostmi se na obrazovce vykreslí (co se týče barvy, nasvětlení, etc.) . V Unity je možné využívat jak shadery napsané ručně, tak je tvořit bez nutnosti programování za pomoci tzv. Shader grafu (ukázka v obrázku 2.1) a Shader objektů. Takto vytvořený shader lze následně selektivně aplikovat na texturu objektu a tím ji upravovat dle podnětů od uživatele.



**Obrázek 2.1.** Ukázka Shader grafu shaderu Paintable z projektu Splatoon's Ink System.

Využití shaderu vlastnoručně vytvořeného, veřejně dostupného hotového shaderu či shaderu implementovaného v rámci uceleného projektu pro vytváření efektu kreslení na papír, hlavně štětcem, ve virtuální-realitě se ukázala být jednou z vhodných možností, jejichž implementaci budu dále popisovat v následující sekci 3.

### 2.1.3 Využití veřejně dostupných metod

Průzkum veřejně dostupných metod proběhl v podobě internetového vyhledávání, jak využitelných projektů či aplikací vytvořených v Unity, tak přímých implementací kreslení na papír v rámci vlastních aplikací napsaných ve variantách a rozšířeních jazyka C (tedy i C# nebo C++), které by mohly být využity pro tvorbu C# skriptů v Unity.

Bohužel veškeré nalezené přímo programované open-sourcové (plně a bezplatně veřejně dostupné) aplikace se ukázaly být nevyužitelné pro toto zadání, jelikož se zabývaly spíše přímou možností kreslení barvami bez dalších efektů, byly tedy pro účely vytvoření efektu kreslení štětce či inkoustového pera nevhodné.

Průzkum dostupných implementací kreslení přímo v Unity již ale poskytl i dva konkrétní využitelné výsledky. Těmi byl projekt vytvořený pro kreslení fixami ve 2D na objekty obsahující texturu v Unity scéně a technické demo pro hru inspirovanou konzolovou hrou Splatoon, jejíž klíčovou mechanikou je možnost kreslení na veškeré dostupné povrchy barvou. Toto jsou dva projekty, které jsem se rozhodl dále upravovat a využít ve vlastní implementaci.

## 2.2 Řezání papíru

Po průzkumu možností kreslení proběhl i krátký průzkum zaměřený na řezání papíru. Tento průzkum proběhl časově až po prototypování a dokonce i zakomponování techniky pro kreslení do virtuálního zážitku. Součástí průzkumu byl i návrh, zda by kvůli výpočetní složitosti nebylo lepší tuto funkcionalitu implementovat bez nutnosti skutečně rozdělovat herní objekt na vícero kusů, ale například za pomoci tvorby nových objektů, které by byly zmenšenou verzí původního papíru a jen by převzaly části textury odpovídající jejich pozici v původním objektu. Tento postup byl však zavrhnut, jelikož by nebyl vhodný pro tvorbu řezů, jejichž výsledkem by nebyl pravoúhlý čtyřúhelník.

Jako nejspokladnější se z časových důvodů se nakonec ukázalo využití již existujícího veřejně dostupného projektu obsahujícího metody pro řezání sítí (mesh neboli polygonální síť je soubor vrcholů, hran a stran (face), které společně určují tvar objektu) objektů v Unity za běhu aplikace (runtime).

# Kapitola 3

## Návrh implementace

Tato kapitola slouží k analýze řešení implementace dvou veřejně dostupných projektů, které jsem se rozhodl využít pro kreslení, návrh zážitku jako takového a analýzu projektu, který bude využit pro řezání papíru.

Součástí začátku návrhu bylo rozhodnout o způsobu implementace virtuální reality jako takové. Za tímto účelem navrhuji využití Unity knihovny pro virtuální realitu SteamVR[7], která umožní plně se zaměřit na implementaci požadovaných funkcionalit projektu, bez nutnosti vlastní implementace rozhraní pro virtuální realitu, sledování kontrolerů a headsetu a dalších funkcionalit nutných pro využití virtuální reality. Jako další možnosti pro implementaci virtuální reality se nabízelo využití XR toolkitu obsaženého v samotném Unity, nicméně se jednalo o toolkit (soubor nástrojů), se kterým jsem na rozdíl od SteamVR knihovny neměl zkušenosti a tak jsem se rozhodl využívat raději knihovnu SteamVR.

Návrh implementace bude vždy v rámci prototypování oddělený, a bude stavět na využití veřejně dostupných GitHubových projektů, se kterými se v rámci této práce dále seznámím a následně je rozšířím o veškeré funkcionality, které v základu chybí. V následné implementaci do uceleného zážitku dojde ke spojení těchto projektů do jednoho, kde každý bude využíván pro jiné účely - jeden projekt bude pro kresbu štětcem, jeden pro kresbu perem a jeden pro řezání papíru. Tyto projekty se jmenují „DokoDemoPainter“[8] a „Splatoon’s Ink System - Mix and Jam“[9] pro kreslení a projekt „Ezy-Slice“[10] pro řezání papíru přičemž všechny tyto projekty jsou veřejně dostupné pod licencí MIT .

### 3.1 Analýza projektu DokoDemoPainter

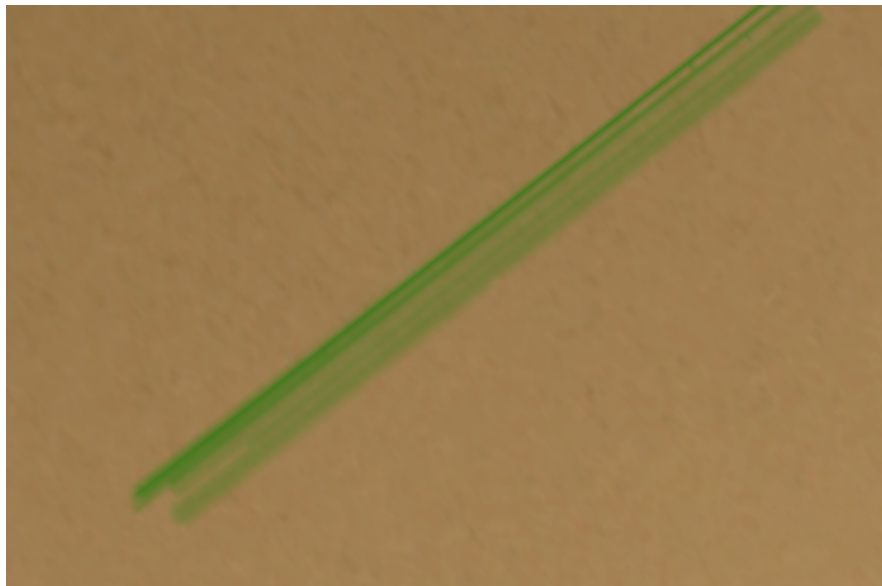
Projekt DokoDemoPainter je souborem skriptů a shaderů pro Unity3D, dovolujících kreslit do textur objektů a zahrnuje funkcionalitu pera, gumy, razítka a možnosti postupného blednutí barvy. Z tohoto projektu plánuji využít funkcionalit kreslení perem a jeho mazání.

Funkce kreslení je v tomto projektu implementována následujícím způsobem: Objekt, který funguje jakožto štětec obsahuje komponentu „Doko Demo Painter Pen“. V této komponentě jsou obsaženy jak parametry štětce, tak i kamera, která musí být v rámci objektu nejlépe jako potomek, a být umístěna aby střed jejího záběru směřoval tam, kde má dojít ke kresbě. Ve chvíli kdy má dojít ke kresbě se za pomoci kamery získá objekt, který je ve středu jejího záběru (tedy objekt, na který se má kreslit), pokud má tento objekt komponentu „Doko Demo Painter Paintable“ (komponenta obsahující metody pro samotné kreslení) a texturu.

Následně dojde k zakreslení do textury výše zmíněného objektu, dle parametrů štětce za pomoci metody obsažené v „Doko Demo Painter Paintable“, která funguje jako shader a zakreslí do textury. Zároveň si objekt pera v rámci této textury zapamatuje kde a zda v ní došlo ke kreslení a v případě opakující se kresby na stejné místo

dojde k vrstvení barvy nebo k odebrání vrstvy, pokud je namísto kreslení zapnuto mazání.

Díky možnosti vrstvit barvu (ukázka 3.1) , kterou tento projekt poskytuje bude využít pro kreslení štětcem ve finálním zážitku.



**Obrázek 3.1.** Ukázka vzhledu vrstvených barev projektu Doko Demo Painter

Komponenta v rámci Unity znamená cokoliv, co je součástí herního objektu. Může se tedy jednat o skripty, kolizní schránky, renderery, materiály atd.

### ■ 3.1.1 Návrh štětce

V tomto projektu jsou již implementovány základní funkce kreslení na papír a možnosti zvětšení šířky štětce či jeho barvy za pomoci skriptů obsažených v tomto projektu. Vstupem pro ovládání kreslení je myš. To pro potřeby tohoto projektu není vyhovující, bude tedy nutné ovládání změnit, aby využívalo kontrolerů pro virtuální realitu.

### ■ 3.1.2 Návrh škálování

Škálování papíru (nebo také měnění velikosti papíru v určitém poměru) je v tomto projektu možné dosáhnout pomocí rovnoměrného zvětšování objektu, na který uživatel kreslí. Tím ovšem dochází i ke zvětšování šířky štětce, která je udávána v pixelech, tedy při změnách velikosti pokreslovaného objektu se bude měnit také (jelikož dochází pouze k „roztažení“ objektu, počty pixelů se nemění). Je tedy třeba tuto šířku upravovat v opačném poměru ke zvětšení tohoto objektu.

### ■ 3.1.3 Návrh kreslení s vodítky

Pro implementaci kreslení pomocí vodítek bylo testováno několik metod, jednou z nich byla i simulace fyzického pravítka v Unity, které by omezovalo kreslicí plochu - simulací hmotného pravítka. Takovéto řešení ale bylo neintuitivní a nedosahovalo žádaných výsledků. Návrh kreslení s vodítky v rámci proratypování bude nakonec spočívat na základě využití virtuálního pravítka, díky kterému bude uživatel schopen udělat rovnou čáru. Virtuální pravítko půjde vypnout a zapnout. V případě zapnutého virtuál-

ního pravítka nebude docházet ke kreslení na papír, ale uživatel bude pouze vybírat počáteční a koncový bod, mezi kterými se nakreslí úsečka.

## 3.2 Analýza projektu Splatooon's Ink System— Mix and Jam

Tento projekt slouží jako technické demo s názornou implementací hlavní mechaniky konzolové hry Splatooon, tedy možnosti střelbou barvy obarvovat herní prostředí. Součástí tohoto projektu je mnoho scén, obsahujících prototypy a i omezenou implementaci herní mechaniky hry „Splatooon“, nicméně tyto scény jsou již příliš specifické pro využití v této práci. Projekt obsahuje ale i scénu obsahující pouze obarvitelné objekty a možnost simulovat myší střelbu barvy. Tato scéna je pro účely projektu již využitelná. Technika kreslení v tomto projektu má jednu hlavní vlastnost a tou je „rozpíjení“ barvy do papíru - díky této vlastnosti tvoří věrohodný efekt inkoustového pera, tedy ve finálním zážitku bude tento projekt využit ke kreslení inkoustovým perem.

Funkčně je podobný předchozímu programu, až na několik klíčových detailů. V základu projekt neobsahuje žádnou kameru, která by určovala, kam se bude kreslit - místo kresby je určené pouze myší a její pozicí na obrazovce (za pomoci Unity funkce hlavní kamery ScreenPointToRay[11], která z místa na obrazovce vyšle relativně vzhledem ke kameře paprsek, stejně jako Raycast), kde dojde ke kontrole, zda je myš na objektu ve scéně, který obsahuje komponentu „Paintable“ (komponenta zajišťující předávání informací o textuře) a následný proces kreslení probíhá v komponentě „Paint Manager“. Dalšími rozdíly je, že tento projekt má vlastní shader program vytvořený v Shader grafu jménem „Paintable“ (skutečně se jedná o název duplicitní s komponentou „Paintable“, tento shader graf byl již použit jako příklad v 2.1) a vyžaduje, aby materiály objektů, na které má být možno kreslit, byly vykreslovány tímto shaderem. Posledním rozdílem je, že tento projekt nemění přímo hlavní texturu materiálu (texturu určující barvy), ale veškeré změny dělá do masky textury, která se vykreslí přes texturu hlavní.

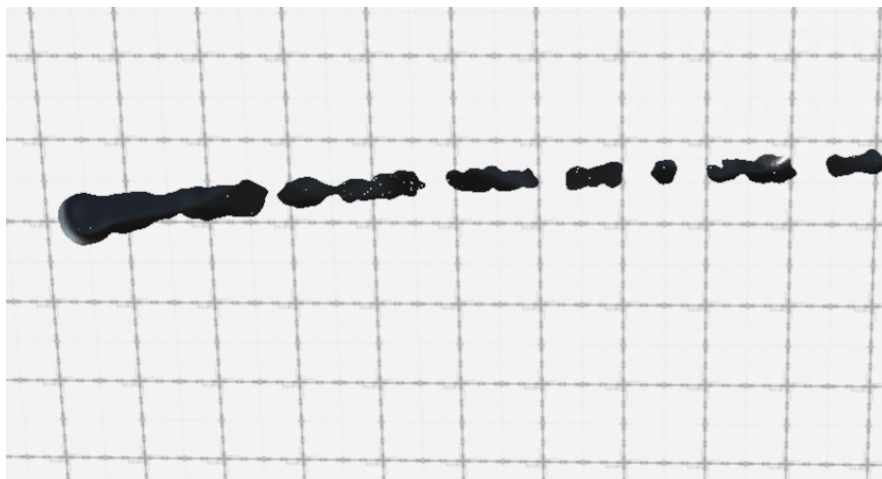
### 3.2.1 Návrh štětce

Projekt využívá shaderů a skriptů k zakreslování přímo do textur (konkrétně masky textur) objektů ve scéně, k čemuž pracuje i s metodou Unity Raycastu[12]. Funkce kreslení je zde tím pádem implementovaná a lze ji v upravené verzi využívat pro virtuální realitu, kde místo ovládání myší bude ovládání za pomoci ovladačů.

Unity Raycast neboli vyslání paprsku funguje tak, jak jeho název napovídá. Vyšle paprsek z bodu ve scéně určeného jako počátek. Paprsek se pohybuje v určeném směru po definovanou vzdálenost a pokud během své cesty narazí na objekt, který má kolizní schránku, vrátí informace o objektu a místě ve scéně, kde k nárazu paprsku do objektu došlo. Tato metoda může vracet i další podrobnější informace, ale pro účely této práce jsou hlavní právě informace o pozici a objektu, se kterým se paprsek setkal.

Kreslení v tomto projektu bude ale potřebovat kromě implementace kontrolerů namísto myši i další úpravu, jelikož je vázáno na Update cyklus Unity, který se spouští vždy jednou za snímek, tedy v případě rychlých pohybů dochází k segmentaci kreslení (příklad segmentace na obr. 3.2).





**Obrázek 3.2.** Ukázka segmentace čar při rychlých tazích při kreslení

Úpravy šířky štětce lze také docílit za pomoci skriptů, které projekt obsahuje. Změny barev jsou možné také, ale podoba, v jaké jsou v základu projektu je nevyhovující našim účelům, jelikož se jedná o barvy, které mají simulovat velké nánosy hutné barvy. Bylo zapotřebí upravit samotné shadery a zbavit se efektů odlesků a efektu prostorovosti, simulujícího hrudky barvy a naopak přidat možnost zprůhlednění barvy za pomoci alfa kanálu barvy.

### ■ 3.2.2 Návrh škálování

Škálování textur a objektu, na který uživatel kreslí lze docílit opět rovnoměrným zvětšováním pokreslovaného objektu a díky řešení šířky štětce, která nevyužívá určování velikosti za pomoci pixelů, dochází rovnou i ke škálování šířky štětce bez nutnosti zasáhnout do skriptu.

### ■ 3.2.3 Návrh kreslení s vodítky

Implementace kreslení s vodítky by měla proběhnout stejně jako v projektu prvním. Součástí implementace této funkce bude v obou případech i vizualizace virtuálního pravítka, tedy úsečka určující odkud a kam následná rovná čára povede.

## ■ 3.3 Ezy-Slice

Tento projekt obsahuje základní metody, kterými lze „rozřezat“ polygonální síť objektu za běhu aplikace v Unity. Projekt EzySlice umožňuje rozříznout polygonální síť objektu rovinou tím, že podél roviny řezu vytvoří na každé straně roviny nové vrcholy, hrany a strany sítě, které jsou oddělené od hran a vrcholů na protější straně roviny. Dále vytvoří dva nové herní objekty a každému z nich přidělí polygonální síť, která se nachází na jedné ze stran roviny řezu a následně původní objekt deaktivuje. Řez za pomoci roviny bude funkcionalitou, která bude implementována do finálního zážitku. Tam bude uživatel moci za pomoci perořízku vybrat dva body, kterými bude procházet rovina, pomocí které dojde k rozdělení papíru na dvě nové části. Funkci pro řezání bude třeba upravit, jelikož objekty vznikající v důsledku řezání neobsahují žádné z komponent původního objektu a není na ně tedy možné následně dále kreslit, či s nimi nějak interagovat ve virtuální realitě.

### 3.4 Návrh zážitku

Virtuálně-realistní zážitek jako takový bude stavět na funkcionalitách navržených výše. Uživatel bude sedět ve scéně Langweilova ateliéru (obr. 3.3) před rýsovacím stolem. Na stole bude k dispozici jako inspirace kus Langweilova modelu s chybějící stěnou. Dále bude na stole papír, který bude sloužit jakožto jediný objekt ve scéně, na který bude možno kreslit. Vedle papíru budou vyskládány kreslicí nástroje, tedy pero, tužka, štětky různých rozměrů, paleta sloužící k výběru barvy, kádinka s vodou pro rozředění a zesvětlení barvy a hadr fungující jako guma. Dále bude na stole pravítko, kterým se bude přepínat mezi kresbou volnou a kresbou s vodítky a perořízek, kterým uživatel bude moci následně kus papíru vyříznout a vložit do modelu pro porovnání. Výřez se bude umět přeškálovat, aby pasoval do modelu.

Uživatel bude ve scéně interagovat s předměty za pomoci speciálně vyvinutého VR ovladače, který by rozměry i možnostmi interakce simuloval kreslicí nástroje. Kreslicí nástroje budou implementovány na stejných principech jak bylo zmíněno výše (viz 3.2.1), jen s přidaným omezením, že ke kresbě bude docházet pouze v případě malé vzdálenosti od papíru. Bylo by ideální implementovat možnost modifikace šířky štětce na základě vzdálenosti od papíru, pro simulaci kresby špičkou, či kresby, když uživatel na štětec tlačí.

Takto bude mít uživatel možnost v atmosférickém prostředí vyzkoušet nakreslit část modelu za pomoci prostředků podobných jako měl Antonín Langweil. Následně bude moci svůj výtvar vložit do modelu pro přímé porovnání.



**Obrázek 3.3.** Scéna „Půda“ ze zážitku „Tvorba Langweilova modelu Prahy“ vytvořená Lukášem Pospíšilem

# Kapitola 4

## Implementace

V této kapitole popíšeme proces implementace řešení navrhnutých v předchozí kapitole 3. Tato kapitola a implementace samotná byla rozdělena na více částí. První jsou implementace prototypů projektů sloužících ke kreslení, pak následuje implementace samotného virtuálně-realistního zážitku.

### 4.1 Implementace DokoDemoPainter

#### 4.1.1 Implementace štěce

První částí implementace bylo prověření možností nastavení štětce, za pomoci skriptu v projektu obsaženém. Tento skript umožňuje nastavit barvu, šířku, průhlednost a tvrdost štětce.

#### 4.1.2 Virtuální realita

Po vyzkoušení těchto funkcionalit proběhl převod projektu do virtuální reality, za pomoci stažení a následného importování knihovny SteamVR do projektu. Z této knihovny byl následně do testovací scény vložen herní objekt Player, obsahující všechny nutné komponenty pro spuštění projektu ve virtuální realitě.

Player neboli hráč, je herní prefab (objekt, který byl vytvořen a uložen a funguje v podstatě jako šablona pro objekt, který lze pouze přidat do scény a je nastaven dle prefabu) knihovny SteamVR, který zajišťuje trackování (sledování) hlavy (headset), rukou (ovladače) a pozici hráče ve scéně (poskytuje i další funkce, ale pro účely práce jsou hlavní právě tyto).

Kreslení ve virtuální-realitě těmito metodami způsobovalo kreslení segmentovaných čar, jelikož kreslicí cyklus byl volán v Update cyklu - tedy jednou za každý snímek, což při rychlejších pohybech způsobovalo segmentované kreslení. Tato funkcionalita byla předělána, aby byla volána za pomoci korutiny (coroutine) [13], tedy v kontextu Unity, za pomoci funkce schopné navazovat na svůj předchozí průběh v rámci po sobě jdoucích snímků. Místo toho, aby kreslení probíhalo vždy jen a pouze během Update cyklu a došlo k nedokončení čar při rychlých tazích, tak je v každém snímku ukládána pozice vždy na začátku snímku a korutina lineárně interpoluje kresbu mezi pozicí uloženou na začátku předchozího snímku a pozicí uloženou v aktuálním snímku. Toto řešení způsobuje efekt „dokreslování“ kdy uživatel udělá na papíru rychlým pohybem čáru, a vidí, jak ke kresbě dochází s lehkým zpožděním, protože kreslil rychleji, než korutina interpolovala, nicméně tento efekt je v porovnání se segmentací kresby zanedbatelný.

Takto převedený projekt ovšem nešel dobře ovládat, jelikož kontrolní schéma stále záviselo na myši. Implementace ovládání za pomoci virtuálně-realistních ovladačů vy-

žadovala nejdříve namapování tlačítek ovladače a následně tvorbu logiky vysílání Raycastů z ovladače.

Řešením bylo umístění herního objektu kamery jakožto potomka ovladače, kterým se bude kreslení ovládat (kameru, která již v tomto projektu byla v ovladači/štětci umístěna kvůli funkci skriptů popsaných v analýze a návrhu 3.1, nebylo možné využít, jelikož kvůli těmto skriptům je vypnutá, pokud nedochází k vykreslování kresby). Po stisknutí předního triggeru ovladače se ze středu této pomocné kamery vyšle Raycast a na povrch objektu v místě dopadu pak dochází ke kreslení aktuálně nastavenou barvou.

Pro účely testování byl k ovladači přidán „LineRenderer“ [14] vykreslující trasu Raycastu vyslaného z kontroleru, aby měl uživatel přesně znázorněno, kam bude kreslit.

Toto řešení s kamerou je využitelné i pro případné kreslení virtuálně-reálnými objekty, kdy jen dojde k úpravě v podobě odstranění podmínky stisknutého triggeru a omezení kreslení jen na blízké okolí konce tohoto objektu.

### 4.1.3 Kreslení s vodítky

Součástí implementace kreslení s vodítky byly pokusy o využití další kamery, která by interpolovala od počátečního místa zacílení kamery v kontroleru ke koncovému místu zacílení a tím by vytvořila rovnou čáru. Toto řešení se ukázalo být slepou uličkou. Ovladače sledované skrze SteamVR objekt hráče z nějakého důvodu nedokáže předat informace o jejich rotaci v herních světových souřadnicích, což byla informace která by určovala odkud, kam a jakým způsobem má kamera interpolovat.

Finálním řešením byla tvorba nového objektu štětce, se stejnými parametry, jako má štětec pro volné kreslení, vlastního skriptu a kamery, posunuté přímo nad samotné plátno/papír a nového skriptu. Funkcionalita skriptu je zjištění počátečního a koncového bodu zamýšlené úsečky a následná lineární interpolace objektu štětce za pomoci korutiny kamery mezi těmito souřadnicemi.

Pro přehlednost byla implementována i vizualizace úsečky za pomoci LineRenderu vykresleného mezi počátečním a koncovým bodem úsečky, určující odkud a kam následná rovná čára povede.



Obrázek 4.1. Ukázka vizualizace kreslení s vodítky a následná rovná čára.

### 4.1.4 Možnost škálování textury

Škálování pokreslovaného objektu a tedy i jeho textury bylo implementováno za pomoci atributů „Scale“ (škála) daného objektu a ovládáno je trackpadem ovladače, využívaným jako D-pad.

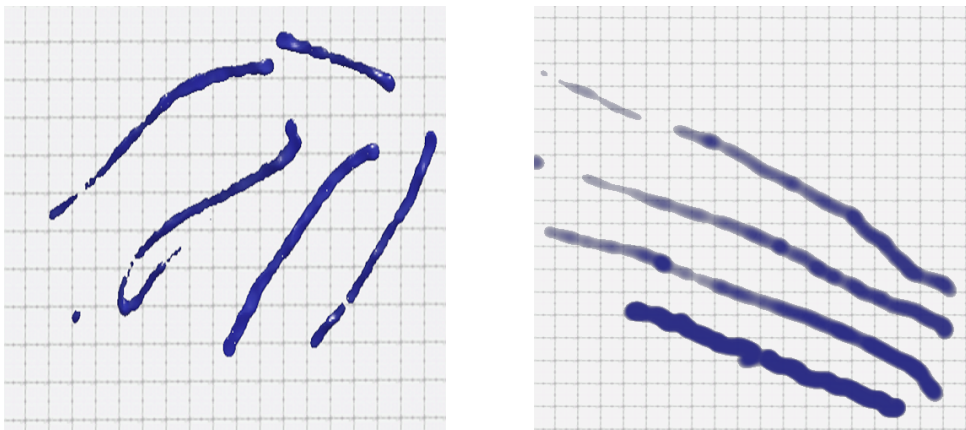
Pro účely testování bylo přidáno i ovládání měnění barev za pomoci klávesnice za běhu projektu - tato funkcionality bude v finálním zážitku nahrazena výběrem barev za pomoci palety.

## 4.2 Implementace Splatoon's Ink System— Mix and Jam

Implementace tohoto projektu proběhla v mnoha ohledech podobně, jako implementace projektu předchozího, budu se tedy soustředit hlavně na rozdíly implementací.

### 4.2.1 Implementace štěce

Po ověření základních funkcionalit v projektu již obsaženého kreslení (nastavení štěce obsahující možnost pro změnu barvy, šířky a tvrdosti) bylo třeba provést úpravu Shader grafu, jak bylo nastíněno v návrhu implementace (viz kapitola 3). Proběhlo tedy odstranění veškerých částí generujících šum, který byl následně využíván pro hrboilaté (bump maps) a normálové mapy, naopak přibylo pronásobení kreslené barvy jejím alfa kanálem, aby bylo možné kreslit tenké, průhledné vrstvy barvy (porovnání vzhledu barvy před a po je vidět na obrázku 4.2).



**Obrázek 4.2.** Porovnání kreslené barvy. Vlevo před úpravou, vpravo po úpravě.

Následně bylo třeba převést hlavní cyklus kreslení z Update cyklu do korutiny (viz 4.1.3), obdobně jako u předchozího projektu. Takto upravený cyklus již netvoří segmentované čáry, ale čáry souvislé, nezávisle na snímkové frekvenci aplikace.

### 4.2.2 Virtuální realita

Tento projekt ovšem způsoboval problémy s implementací SteamVR knihovny, jelikož nezobrazoval výstup z kamer SteamVR hráče do headsetu a zobrazoval pouze černou a bílou barvu v displejích headsetu. Po konzultaci s vedoucím práce bylo doporučeno vyzkoušet projekt převést do projektu, který by začínal s odlišným nastavením Unity shaderů, jelikož změny nastavení současného projektu tento problém nevyřešily. Problém byl vyřešen až převedením celého prototypu do projektu využívajícího Universal Render Pipeline.

Universal Render Pipeline, dále jen URP, je jedním z vykreslovacích potrubí (pipeline) poskytovaných Unity, které se liší od ostatních hlavně lépe optimalizovaným vykreslováním světel. Stínování světel je v této pipeline vykreslováno opožděně (deferred rendering), což znamená že k němu dochází až v samostatném průchodu vykreslování (render pass) poté, co jsou vykresleny všechny vrcholové a pixelové shadery (vertex & pixel shader). Tím se oddělí výpočty geometrie scény od výpočtů světel a dojde ke zlepšení výkonu tím, že se vypočítávají světla pouze ve viditelných částech scény.

Vzhledem k ovládání, které již využívalo raycastů, došlo jen k nahrazení myši jako jejich zdroje za kameru opět umístěnou ve virtuálně-reálném ovladači.

### ■ 4.2.3 Kreslení s vodítky

Implementace kreslení dle vodítek opět proběhla za využití korutiny, ale bez potřeby vytváření další kamery či nové logiky. Bylo využito metody pro volné kreslení, která byla volána na interpolované body mezi počátečním a koncovým bodem kreslené úsečky. Vzhledem k implementaci kreslení za pomoci shaderů, které dokázalo simulovat i efekt rozpíjené barvy, když uživatel kreslí pomalu, docházelo i k opaku, tedy tenkým čarám, když uživatel kreslí rychle. Jelikož původní implementace kreslení rovných čar měla pevně danou dobu korutiny, docházelo k různé tloušťce čar v závislosti na její délce. Tento problém byl vyřešen přidáním výpočtu pro dobu korutiny v závislosti na vzdálenosti počátečního a koncového bodu ve světových souřadnicích v Unity.

### ■ 4.2.4 Možnost škálování textury

Implementace škálování proběhla stejně jako v předchozím projektu a i do tohoto projektu byly přidány veškeré funkcionality za účelem testování, jakými je vizualizace kreslení s vodítky, vizualizace místa, kam uživatel kreslí a možnost nastavení barvy a její průhlednosti za běhu projektu.

## ■ 4.3 Implementace do uceleného zážitku „Tvorba Langweilova modelu Prahy“

Tato podkapitola implementace obsahuje popis jak jsem postupoval během implementace prototypů navržených a vytvořených výše (4.1 a 4.2) do již existující scény zážitku „Tvorba Langweilova modelu Prahy“. Podkapitola je rozdělena do sekcí, každá sekce se věnuje jedné z interakčních technik.

### ■ 4.3.1 Scéna

Scéna jako taková již existovala a byla vytvořena v rámci již několikrát zmíněného zážitku „Tvorba Langweilova modelu Prahy“. Z tohoto zážitku jsem využil scénu „Půda“ (vytvořenou Lukášem Pospíšilem), jelikož rozdíly mezi scénami obsaženými v existujícím zážitku byly hlavně v jejích skriptech, z nichž jsem ponechal a využil pouze jeden a to skript na otáčení podstavky modelu, díky kterému si může uživatel lépe prohlédnout model, který má jako předlohu.

Tato scéna ovšem nešla využít bez úprav, jelikož byla vytvořena ve starší verzi Unity editoru (verze 2019.2.9) a materiály v ní obsažené nevyužívaly URP shadery, které jsem využíval již během prototypování. Bylo třeba tedy celou scénu převést do

novější verze Unity (verze 2020.3.21) a její materiály převést na materiály využívající URP shader.

Některé materiály obsažené v knihovně SteamVR, bylo třeba převést na URP shader verze manuálně, jelikož automatické převedení Unity tyto materiály zcela vynechávalo. Věcí, která se nepovedla do URP převést vůbec byly billboardy (2D textury, které se otáčejí tak, aby se otáčejí v závislosti na pohledu uživatele) využívané na nejvzdálenější knihovny v scéně dílny. Po krátké úvaze byly textury billboardů nahrazeny úplně novými texturami, které simulují knihovny otočené zadní přepážkou k uživateli.

Poté, co scéna již měla všechny potřebné materiály, bylo třeba ještě upravit její osvětlení. Množství a intenzita světla ve scéně způsobovaly ve virtuální realitě pře-světlení objektů na stole, na kterém má uživatel pracovat a zkreslovalo barvy, jelikož osvětlení simulovalo žluté světlo lamp a svíček. Celkově byla intenzita světla snížena a jejich množství zredukováno. Toto bylo pravděpodobně způsobeno přechodem na URP shadery, jelikož v této vykreslovací pipeline došlo ke změnám výpočtů osvětlení (viz sekce 4.2.2).



**Obrázek 4.3.** Porovnání osvětlení v ateliéru před a po úpravě.

### 4.3.2 Kreslení

Implementace kreslení nakonec proběhla jinak, než bylo zpočátku plánováno. původním záměrem byla tvorba dvou prototypů, z nichž dojde k vybrání jednoho, který bude implementován do zážitku a dále rozvíjen. Místo tohoto postupu došlo k rozhodnutí ponechat oba prototypy a využít každý pro něco jiného - tedy

i implementaci kreslení bylo třeba provést dvojí, pro každý prototyp jednou.

Pro kreslení perem bylo rozhodnuto o využití prototypu založeném na projektu Splattoon's Ink System— Mix and Jam (viz 4.2), jak již bylo dříve zmíněno, kvůli efektu rozpíjejícího se inkoustu. Veškeré skripty byly tedy importovány z prototypu a byla přetvořena logika pro kreslení. Prototypy využívaly k ovládní přímo VR ovladače, kterými uživatel ukazoval kam chce kreslit a kreslil stisknutím triggeru ovladače. Tato logika byla přetvořena, aby zdrojem kresby a s ním asociovaných Raycastů byla v tomto případě špička inkoustového pera, které může uživatel uchopit jak uzná za vhodné a následně s ním kreslit tím, že ho dostatečně přiblíží k papíru. Pro lepší uživatelskou přívětivost je tato vzdálenost indikována poloprůhlednou čarou vykreslenou do hrotu pera, která když se dotkne papíru, tak značí, že je pero dostatečně blízko aby došlo ke kresbě.

Pro kreslení s vodítky byl implementován přepínač, jakožto skript, který má objekt pravítka na stole v ateliéru - při dotknutí se pravítka perem dojde ke zvýraznění pravítka a aktivaci kreslení rovných čar. Uživatel určuje kde se rovná čára nakreslí tím, že při zapnutém kreslení rovných čar dostatečně přiblíží pero k papíru, tímto prvním „dotekem“ určí počáteční bod úsečky. Koncový bod úsečky je poslední pozice pera předtím, než ho uživatel oddálí od papíru. Po oddálení pera od papíru dojde k nakreslení rovné čáry. Během celého tohoto procesu má uživatel vizualizovanou podobu rovné čáry kontrastní rudou čarou vykreslenou LineRenderem, znázorňující kudy bude výsledná úsečka procházet.

Toto jsou funkcionality inkoustového pera ve finální implementaci - jedná se o černé inkoustové pero, které má konstantní tloušťku kresby, nelze mu měnit barva a lze s ním provádet kresbu pomocí vodítek, která se přepíná pravítkem umístěným ve scéně. Tímto jsou splněny funkční požadavky FP1, FP7 a FP8 (viz podkapitola 1.4.1).

Na kreslení štětcem byl využit druhý prototyp založený na projektu DokoDemoPainter (viz 4.1). Implementace započala prvotními potížemi s faktem, že shader materiálu papíru nutný pro funkci projektu Splattoon postrádal správně označenou hlavní texturu. Po úpravě shaderu, aby s materiály jím vytvořenými dokázal pracovat i tento prototyp již ale implementace probíhala identicky s prototypem pro kresbu perem. Následně proběhlo přidání mechaniky, kdy se tloušťka kresby odvíjí od vzdálenosti zdroje Raycast paprsku k papíru, tedy když bude uživatel kreslit „špičkou“ štětce, kresba bude tenčí, než když bude kreslit štětcem na který „tlačí“.

Pro kreslení štětcem kresba s vodítky nebyla implementována, bylo usouzeno, že kreslení „podle pravítka“ v případě malby štětcem nedává z hlediska věrnosti smysl. Naopak kvůli věrnosti, byla pro malbu štětcem přidána možnost výběru barvy, kterou uživatel maluje, za pomoci objektu palety s barvami ve scéně. Uživatel si vybere barvu tím, že se jí štětcem dotkne na paletě, tím se i změní textura samotného štětce a na jeho špičce se zobrazí zrovna vybraná barva, kterou následně může uživatel kreslit. Výběr barev byl implementován za pomoci kolizních schránek na paletě, kde každá schránka určuje jednu barvu. Změna barvy na štětci je implementována mnoha různými texturami pro objekt štětců, kde má každá textura různě obarvenou oblast



špičky štětce.

Po implementaci druhého prototypu má tedy uživatel k dispozici dva štětce různých velikostí a tloušťky kresby, u kterých může měnit barvu kresby skrze paletu s barvami a tloušťku kresby vzdáleností štětce od papíru. Tímto jsou splněny funkční požadavky FP2, FP4 a i opětovně FP7 (viz podkapitola 1.4.1).

### ■ 4.3.3 Mazání

Implementace mazání byla nutná taktéž dvojnásobně, jak z hlediska mechanického, tak věrovnostního. Pro mazání kresby perem byl do scény přidán objekt bílo-modrého mazátka, které je funkčně identické k peru, jen s tím rozdílem, že kresbu perem maže. Mazání probíhá „překreslením“ pera v masce materiálu barvou s alfa kanálem nastaveným na 0, tedy pro účely Unity „průhlednou“ barvou. Toto mazání má oproti kresbě zvětšený průměr „kresby“.

Mazání štětce na druhou stranu nevyžadovalo přidání žádného nového objektu do scény. Funkčně mazání probíhá při kresbě štětcem, který na sobě nemá žádnou barvu. V tomto stavu štětce začínají při zapnutí scény a zároveň je možné je do tohoto stavu dostat otřením štětce o hadřík umístěný ve scéně na stole nad papírem. Při otření o hadřík se ze štětce odstraní barva, textura modelu se přepne na texturu bez barvy na špičce štětce a dojde ke změně módu kresby na mazání, kdy místo přidávání dalších vrstev barvy štětcem začne docházet k jejich postupnému odebírání. To také přidává k efektu „věrohodnosti“ kdy prázdný namočený štětec rozředuje barvu na papíře a eventuálně ji rozředí natolik, že již na papíře není vidět. Průměr mazání se řídí stejnými pravidly jako kresba štětcem.

Uživatel má po implementaci k dispozici metody pro mazání jak kresby perem, tak štětcem, které může mazat odděleně od sebe, s pevně daným průměrem u mazání pera a možností tloušťku upravovat u mazání štětcem stejně jako při kresbě. Tímto je splněn funkční požadavek FP11 (viz podkapitola 1.4.1).

### ■ 4.3.4 Řezání

Mechanika řezání neprošla fází prototypování, ale byla přidána až přímo při tvorbě uceleného zážitku. Řezání bylo implementováno obdobně jako kresba s vodítky. Uživatel perořízkem vybere počáteční a koncový bod úsečky, ale je potřeba vypočítat rovinu, podél které řez proběhne. Rovina se v Unity dá jednoznačně určit za pomoci bodu, který na ní leží a vektoru, který je na ní kolmý a určuje její orientaci v prostoru. Máme k dispozici nekonečně mnoho bodů na úsečce mezi počátečním a koncovým bodem. Z těchto údajů dokážeme určit bod B, který na úsečce leží (například počáteční bod přímky) a minimálně dva vektory  $u$  a  $v$ , které rovinu určují.

Výsledkem vektorového součinu vektorů  $u$  a  $v$  je vektor  $w$  kolmý na rovinu určenou vektory  $u$  a  $v$ .

Pro výpočet vektorového součinu existuje v Unity funkce `Vector3.cross` (nebo také `cross product` - vektorový součin). Tímto výpočtem získáme vektor kolmý na vektory  $u$  a  $v$  a máme vše potřebné, pro určení roviny, která vznikne na úsečce mezi počátečním a koncovým bodem, které vybral uživatel a podél které se provede řez.

Jak bylo zmíněno v analýze (viz kapitola 3.3), objekty vzniklé při řezu obsahují jen materiál a komponenty nutné pro vykreslení objektu ve scéně, je tedy třeba skripty pro generování nových objektů při řezu upravit. Nově vzniklé objekty musí obsahovat i komponenty SteamVR (Interactable a Throwable - zajišťují, že se s herními objekty dá manipulovat za pomoci ovladačů a popisují jak se objekt při interakci má chovat) a obou kreslicích prototypů (Paintable a DokoDemoPaintable), které jsou nezbytné pro to, aby se na nové objekty dalo kreslit - tyto objekty jsou stále kusy papíru, se kterými uživatel musí být schopen interagovat a dále na ně v případě potřeby kreslit.

Samotné přidání těchto komponent ovšem nestačí, jelikož při tvorbě nových objektů, které využívají komponentu Paintable (komponenta zajišťující, aby na objekt bylo možné kreslit perem) dojde k vytvoření nové masky pro materiál, tudíž při rozříznutí papíru dojde ke ztrátě veškeré perokresby. Bylo tedy nutné upravit skript zajišťující inicializaci této komponenty, aby novou masku vytvořil pouze v případě originálního, celistvého papíru a ve všech ostatních případech pouze využíval masku původního papíru. Kvůli instancování materiálů bylo třeba provést obdobnou úpravu i pro samotnou texturu materiálu, jelikož ta je využívána k aplikaci kresby štětcem. Při tvorbě objektů vzniklých rozříznutím dojde k přiřazení stejného materiálu, jaký měl původní objekt do objektů nově vzniklých, ovšem kvůli tomu, že se pouze jedná o novou instanci tohoto materiálu, tak jeho textura neobsahuje žádnou z kreseb štětcem, které byly v mezičase vytvořeny. Je tedy třeba texturu materiálu převzít přímo z původního objektu.

V rámci implementace řezání papíru byla přidána i funkcionálita „slepení“ papíru, která umožňuje vrátit papír do podoby, v jaké byl před jeho rozříznutím. V zážitku lze tuto funkcionálitu zavolat dotknutím se tlačítka „Slepit papír“, které se nachází na druhé stránce knihy, která je na stole a obsahuje stručné instrukce pro uživatele. Funkčně je tato funkcionálta implementována tak, že místo deaktivace původního rozřezávaného objektu, pouze dojde k jeho posunutí mimo scénu a za pomoci dvou zásobníků - jednoho pro objekty papíru ve scéně a jednoho pro objekty mimo scénu se udržuje přehled o objektech a při stisknutí tlačítka „Slepit papír“ se odsunou rozřezané kusy papíru a vrátí se předchozí celý kus papíru. Stisknutí tlačítka vždy vrátí poslední řez a lze ho opakovat, dokud na stole není opět celý neporušený papír.

Rozřezávání papíru s sebou nese i omezení, která se nepodařilo vyřešit. Jmenovitě při rozříznutí papíru dojde ke ztrátě informace o vrstvách barev štětce, tedy nic namalované štětcem, nelze již po rozříznutí papíru smazat a při „slepení“ papíru není implementováno přejímání textury materiálu popisované výše. Pokud tedy uživatel papír rozřízne a dále kreslí štětcem na odříznutou část a provede akci „Slepit papír“, tak kresba provedená na odříznutou část papíru zmizí. Bude pouze na kusu odříznutého papíru, který je nyní mimo scénu a na kusu papíru, který se do scény vrátil, chybí.

Uživatel má po implementaci k dispozici perořízek, který může uchopit stejně jako ostatní nástroje a jeho pomocí může na papíře vybrat dva body, skrze které bude vést řez celým papírem. Papír lze takto neomezeně rozřezat a byla implementována i funkce „Slepení papíru“, kterou může uživatel papír znovu sestavit. Tímto jsou splněny funkční požadavky FP5 a opětovně i FP8 (viz podkapitola 1.4.1).

### ■ 4.3.5 Škálování papíru

Principy škálování navržené pro každý prototyp zvlášť bylo třeba také sloučit, ovšem vzhledem k podobnosti této funkcionality mezi projekty k žádným potížím v rámci implementace samotné nedošlo. Komplikace ovšem vyvstaly při implementaci do zážitku samotného, jelikož bylo třeba přijít na způsob, jakým by uživatel mohl využít interaktivně a ne pouze za pomoci nerealistické interakce tlačítkem na ovladači. Po konzultaci s vedoucím práci se rozhodlo pro implementaci, kdy ke škálování dojde ve chvíli, když uživatel uchopí papír oběma rukama a bude se škálovat v poměru, v jakém uživatel začne ruce přibližovat či oddalovat zatímco papír drží. V této formě bylo škálování implementováno.

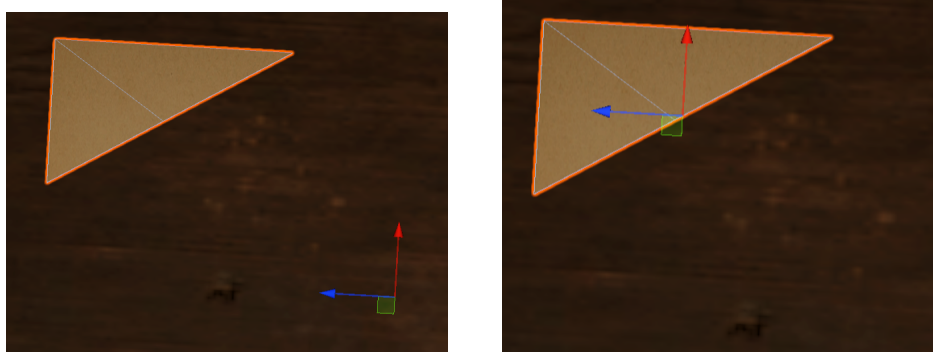
Tato funkcionalita přinesla jedno další omezení, které se v průběhu práce nepodařilo opravit. Tím je, že v případě zvětšeného papíru a zapnutého kreslení za pomoci vodítek uživatel nevidí, že by ke kresbě docházelo. To je způsobeno tím, že při škálování papíru dojde ke zmenšení průměru kresby pera, a to do takové míry, že výsledná rovná linie skoro není vidět. Pokud by se toto škálování nedělo, tak by pero mělo stále stejnou velikost kresby, bez ohledu na velikost papíru a bylo by nemožné za pomoci pera nakreslit tenké linie, kterých lze docílit při volném kreslení a zvětšeném papíru.

Po implementaci má uživatel možnost zvětši si papír až na dvojnásobek původní velikosti nebo ho libovolně zmenšovat. Při zvětšení papíru ovšem velikosti kresby nástrojů zůstávají stejné, tedy kresba perem na zvětšený papír bude mít za následek tenčí čáru, po opětovném zmenšení papíru na původní rozměr. Tímto je splněn funkční požadavek FP10 (viz podkapitola 1.4.1).

### ■ 4.3.6 Vkládání do modelu

Posledním funkčním požadavkem, který bylo třeba implementovat je vkládání výsledné kresby do samotného modelu s chybějící stěnou, který nám v zážitku slouží jako předloha. To probíhá za pomoci kolizní schránky vložené na místo chybějící stěny, která funguje jako spouštěč (Trigger). Ve chvíli, kdy se do této kolizní schránky vloží kus papíru, ať už ořezaný či neořezaný, tak se jeho rozměry nastaví na původní (přeškáluje se na velikost, jakou měl tento kus papíru při spuštění zážitku) a dojde k jeho vložení na pozici kolizní schránky.

Při implementaci této funkcionality bylo zjištěno, že kusy papíru vzniklé rozříznutím, mají střed objektu (střed pozice komponenty Transform) na stejném místě jako původní papír a to způsobuje vložení na nesprávnou pozici (obr. 4.4). Řešením bylo vypočítání rozdílu mezi pozicí středu vykresleného objektu (parametr komponenty Renderer, jménem bounds.center, neboli střed hranic objektu) a pozicí, kterou má objekt nastavenou v rámci scény. Následně při vložení dojde k posunutí objektu o tento rozdíl.



**Obrázek 4.4.** Porovnání pozic středu komponenty „Transform“ (vlevo) a středu objektu (vpravo)

Model si drží informaci o objektu papíru, který do něj byl uživatelem vložen a udržuje ho na místě stěny i při otáčení modelem, dokud není objekt papíru opět vyjmut. Tímto je splněn funkční požadavek FP9 (viz podkapitola 1.4.1).

# Kapitola 5

## Testování

Tato kapitola popisuje proces testování náhodnými uživateli různých věkových kategorií (převážně 1. a 2. stupeň ZŠ) a testování výkonu implementace zážitku.

### 5.1 Proces testování zážitku

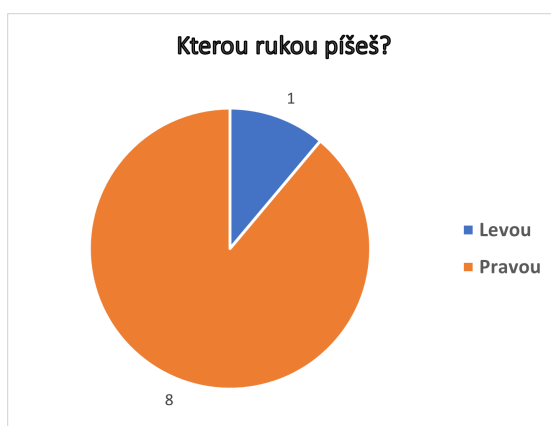
Testování probíhalo formou uživatelského testování jakožto jedna ze součástí naučného programu, který byl připraven během návštěvy ČVUT 11.5.2022 pro žáky ZŠ.

Uživatel byl usazen na židli u stolu a měl na hlavě virtuálně-reální headset a v ruce ovladače s headsetem spárované. Vždy před samotným testováním proběhlo seznámení uživatele s formou a cílem zážitku, seznámení s ovládacími prvky zážitku a následně již 3-5 minutová fáze, kdy si uživatel mohl zážitek vyzkoušet. V závislosti na rychlosti, jakou proběhl úvod, následovalo 10-15 minut, kdy uživatel mohl pracovat na tvorbě kusu Langweilova modelu Prahy ve virtuálně-reálním zážitku vytvořeném v rámci této práce. Po uplynutí tohoto časového úseku či dokončení práce uživatel vyplnil dotazník ohledně zážitku, který právě testoval.

Dotazník A byl vytvořen z návrhů dotazníků pro tuto práci a návrhu dalšího autora práce se stejným zadáním a sestával z otázek s možností zaškrtnout odpověď, volných otázek a otázek s odpovědí v rámci škály od 1 do 5.

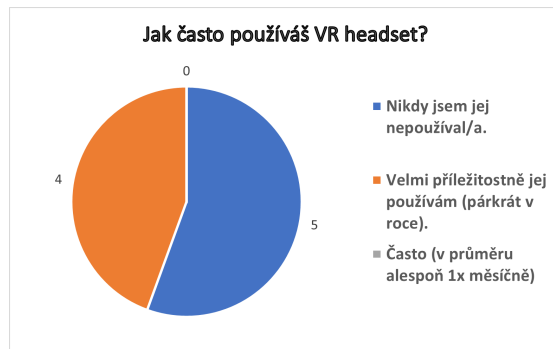
#### 5.1.1 Výsledky z dotazníků v grafech

V této podkapitole jsou zpracovány výsledky z uživatelského testování do grafů s relevantními komentáři.



**Obrázek 5.1.** Graf zobrazující výsledky otázky „Kterou rukou píšeš?“ z dotazníku.

Z grafu (obr. 5.1) je patrné, že na otázku „Kterou rukou píšeš?“ drtivá většina (88%) účastníků uživatelského testování odpověděla, že pravou.



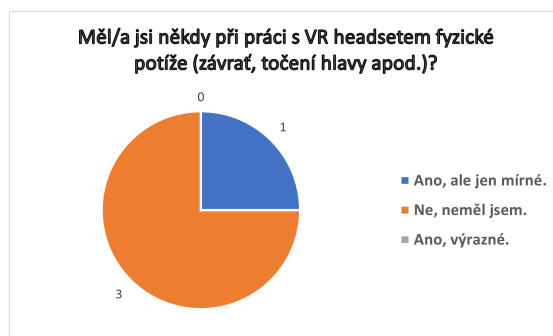
**Obrázek 5.2.** Graf zobrazující výsledky otázky „Jak často používáš headset?“ z dotazníku.

Z odpovědí na otázku „Jak často používáš headset?“ (obr. 5.2) vyplynulo, že pro nadpoloviční většinu (55%) uživatelů bylo toto testování také jejich prvním setkáním s VR headsety obecně.



**Obrázek 5.3.** Graf zobrazující výsledky otázky „Jaký způsob ovládání u headsetu jsi v minulosti používal/a?“ z dotazníku.

Mezi uživateli, kteří již měli předchozí zkušenosti s VR headsety (4 uživatelé) bylo nejrozšířenější (75% odpovědí) metodou ovládání využívání ovladačů (obr. 5.3).



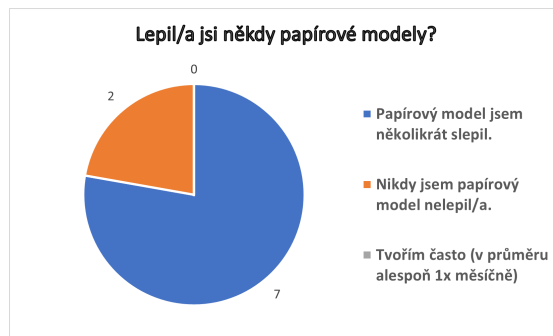
**Obrázek 5.4.** Graf zobrazující výsledky otázky „Měl/a jsi někdy při práci s VR headsetem fyzické potíže?“ z dotazníku.

Z těchto uživatelů (4 uživatelé) také většina (75%) v minulosti buď nepocítovala žádné zdravotní potíže po používání headsetu nebo jen potíže mírné (obr. 5.4).



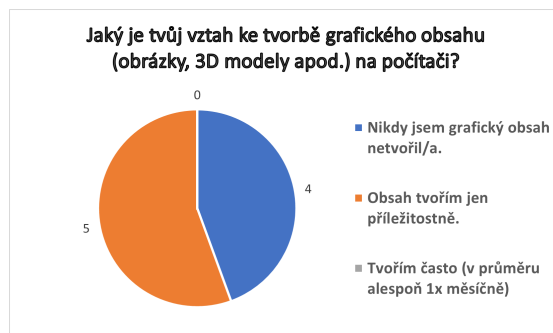
**Obrázek 5.5.** Graf zobrazující výsledky otázky „Jaký je tvůj vztah ke kreslení na papír?“ z dotazníku.

Na otázku „Jaký je tvůj vztah ke kreslení na papír?“ ze všech uživatelů (9 uživatelů) odpověděla nadpoloviční většina (55%), že alespoň jednou do měsíce kreslí a zbytek uživatelů je rozdělen na stejné části (po 22.5%) těch, kteří kreslí příležitostně nebo vůbec (obr. 5.5).



**Obrázek 5.6.** Graf zobrazující výsledky otázky „Lepil/a jsi někdy papírové modely?“ z dotazníku.

Otázka „Lepil/a jsi někdy papírové modely?“ měla více jak dvě třetiny (77%) odpovědí stejné a to, že uživatel už několikrát papírové modely lepil, ale ne nijak pravidelně (obr. 5.6). Velké množství kladných odpovědí mohl mít za následek předcházející program dne, který zahrnoval i lepení papírových modelů.



**Obrázek 5.7.** Graf zobrazující výsledky otázky „Jaký je tvůj vztah ke tvorbě grafického obsahu na počítači?“ z dotazníku.

Další otázkou bylo „Jaký je tvůj vztah ke tvorbě grafického obsahu (obrázky, 3D modely apod.) na počítači?“ Nadpoloviční většina (55%) uživatelů příležitostně tvoří grafický obsah, přičemž zbytek uživatelů grafický obsah nikdy netvořil (obr. 5.7). Na tuto otázku v případě kladné odpovědi navazovala otázka jaký program uživatel k tvorbě využíval. Největší část (2 uživatelé) těchto uživatelů tvoří grafický obsah v Malování a jeden uživatel v aplikaci Photoshop (jeden uživatel neodpověděl a jeden pouze obecně odpověděl „design app“). K ovládání aplikací na tvorbu grafického obsahu uživatelé nejčastěji využívali počítačovou myš (3 uživatelé), či dotykový tablet bez pera (2 uživatelé).

Veškeré otázky, které měly možnost být zodpovězeny za pomoci numerické škály od 1 do 5 byly shrnuty do grafu 5.8. Je možné že výsledky z tohoto grafu nejsou odpovídající skutečnosti. Toto podezření je způsobeno nesrovnalostmi mezi slovním hodnocením v otevřených otázkách a hodnocením na numerické škále u minimálně dvou uživatelů. Mohlo dojít k nepochopení hodnotícího systému uživateli, které mělo za důsledek nesprávné vyplnění tohoto typu otázek.



**Obrázek 5.8.** Graf zobrazující výsledky veškerých otázek, které měly možnost odpovědi na škále od 1 do 5

Hodnocení těchto otázek se obecně podobalo školskému známkovacímu systému, kde 1 je hodnocení nejlepší a 5 nejhorší (uživatel byl s funkcí velmi spokojen - 1 nebo velmi nespokojen - 5). Jediná otázka, která se tímto pravidlem neřídila je otázka druhá (bráno od horní části grafu - názor na rozmanitost nástrojů), kde hodnocení 1 znamenalo, že nástrojů bylo k dispozici málo a 5 naopak, že nástrojů bylo příliš.

Uživatelé v dotazníku také vyplňovali doplňující otevřené otázky, zaměřené přímo na zpětnou vazbu ohledně virtuálního zážitku. z těchto otázek vyplynulo, že po fyzické stránce bylo po testování dobře všem uživatelům kromě dvou - jeden zažíval lehké bolesti hlavy a druhý lehké rozostření zraku.

Kladně hodnocená byla v zážitku nejčastěji funkce řezání papíru, možnosti interakcí s nástroji, samotné kreslení na papír a též grafické zpracování ateliéru.

Záporně zhodnocena byla manipulace s papírem, nesnadnost zvedání předmětů ze stolu, mazání a práce se štětcem, který se nedržel pohodlně.

Zpětnou vazbu v podobě otázky, co uživateli v zážitku chybí, většina vyplnila neutrálně (6 odpovědí bylo „nic“ a „nevím“). Uživateli navržené úpravy byly větší paleta barev, možnost kreslit prsty a upravit ovládání, aby se nástroje držely jako pero.



### 5.1.2 Výsledky uživatelského testování

Uživatelské testování proběhlo na skupině uživatelů, pro které byl všeobecně virtuální zážitek novou zkušeností, stejně jako tvorba grafického obsahu na počítači. Bereme-li v potaz výsledky dotazníku v podobě v jaké jsou (tedy i se zmíněnými domnělými špatnými odpověďmi), tak nejpozitivněji bylo hodnoceno prostředí samotné, které nevzniklo v rámci této práce.

Většinou kladně zhodnocená byla i synchronizace reálného ovladače a ovladače v zážitku, malování štětcem a možnost mazání.

Záporné hodnocení sklídila hlavně interakce s papírem a podobnost projektu s kreslením na papír.

## 5.2 Výkonnostní testování

Výkonnostní testování bylo třeba vztáhnout vzhledem k výkonu původního zážitku „Tvorba Langweilova modelu Prahy“. Došlo tedy k testům jak v neupraveném původním virtuálním zážitku, tak zážitku, který byl výstupem této práce. Testování proběhlo na sestavené verzi scény „Půda“, z projektu „Tvorba Langweilova modelu Prahy“, jelikož se jednalo i o scénu, která byla následně využita v této práci k tvorbě virtuálního zážitku. Testování proběhlo na osobním notebooku, který disponuje procesorem Intel(R) Core(TM) i5-8300H CPU 2.30GHz, grafickou kartou NVIDIA GeForce GTX 1050 a 24GB RAM. Vývoj a prototypování proběhlo na VR headsetu Acer Windows Mixed Reality AH101 s obnovovací frekvencí 90Hz a implementace do uceleného zážitku probíhala na VR headsetu Oculus Quest 1 s obnovovací frekvencí 72Hz a uživatelské testování proběhlo na headsetu Oculus Quest 2 s frekvencí 90Hz. Výkonnostní test jako takový spočíval v 5 minut dlouhém pobytu v zážitku, během kterého se prováděly veškeré interakce, které provádí běžný uživatel, jako je kreslení, rozhlížení ve scéně, otáčení vzorovým modelem a další.

### 5.2.1 Testování projektu „Tvorba Langweilova modelu Prahy“

Kvůli využití slabšího stroje, než využíval autor „Tvorba Langweilova modelu Prahy“, jsou i výsledky výkonnostního testování slabší. Při spuštění sestavené verze s původního projektu je k dispozici možnost výběru kvality grafiky a rozlišení. Jako rozlišení bylo zvoleno nativní rozlišení headsetu oculus Quest 1, tedy 3440 x 1440 pixelů.

Kvalita	Průměr FPS	Minimum FPS	GPU Frametime (ms)	CPU Frametime (ms)
Ultra	27.2	24	38-42	8-12
Very High	29.2	25	32-36	7-11
High	36.3	34	23-28	6-10
Normal	38.3	35	18-21	6-8
Low	57.2	38	13-20	5-10
Very Low	57.5	36	12-17	5-8

**Tabulka 5.1.** Výsledky výkonnostního testování projektu „Tvorba Langweilova modelu Prahy“ v různých nastaveních grafické kvality.

Tento projekt byl co se výkonu týče stabilní, s nepatrným poklesem výkonu ve chvíli, kdy docházelo ke kreslení samotnému.

### 5.2.2 Testování aktuálního zážitku

Tento projekt po sestavení nenabízí volbu grafické kvality, ale je vždy sestaven v určité kvalitě. Testování proběhlo tedy na každé z defaultních nastavení pro grafickou kvalitu, které poskytuje URP projekt v Unity - *High*, *Medium* a *Low*. Jako rozlišení bylo zvoleno nativní rozlišení headsetu oculus Quest 1, tedy 3440 x 1440 pixelů.

Kvalita	Průměr FPS	Minimum FPS	GPU Frametime (ms)	CPU Frametime (ms)
High	54.2	32	10-18	6-14
Normal	67.3	36	10-12	5-8
Low	68.7	40	5-8	3-6

**Tabulka 5.2.** Výsledky výkonnostního testování projektu „2D kreslení na papír ve virtuální realitě“ v různých nastaveních grafické kvality.

Oproti původnímu projektu došlo v této verzi ke snížení počtu zapečených světél ve scéně, což vedlo už v průběhu vývoje ke zlepšení výkonu, přičemž díky změně chování intenzity světla v URP projektu je ve scéně stále dostatek světla, aby mohl uživatel pohodlně pracovat na tvorbě modelu. V tabulce uvedené hodnoty jsou průměrnými hodnotami, které se v rámci testování vyskytovaly, nicméně bylo zjištěno, že volná kresba perem při vyšších kvalitách drasticky snižuje FPS (Frames Per Second - snímky za vteřinu) - řádově o 10-15 FPS a řezání papíru je výpočetně náročné, tedy chvilkově zhorší odezvu CPU na 45ms, nicméně se výkon projektu s dobou běhu nezhoršuje a zůstává konstantní.

### 5.2.3 Výsledky výkonnostního testování

V rámci porovnání byla všechna nastavení grafické kvality upravena, aby co nejvíce odpovídala nastavením, původního virtuálního zážitku - tedy nastavení *High*, *Medium* a *Low* si mezi projekty odpovídají. Z výsledků testování je patrné, že přechod na projekt využívající URP shader a snížení počtu zapečených světél pozitivně ovlivnilo výkon a i na slabším stroji lze dosáhnout při nižších nastaveních kvality relativně stabilní snímkové frekvence, která se blíží obnovovací frekvenci headsetu, tímto je splněn funkční požadavek FP13 a nefunkční požadavek NFP2 (viz podkapitola 1.4.2). Projekt by ovšem bylo možné dále optimalizovat, například aby nedocházelo k tak výraznému poklesu výkonu při volném kreslení perem.

# Kapitola 6

## Závěr

V této kapitole proběhne zhodnocení práce v kontextu splnění požadavků implementace a také z pohledu uživatelského testování.

### 6.1 Požadavky implementace

Požadavky popsané v kapitole 1.4 byly úspěšně implementovány, s výjimkou požadavku FP3 (viz podkapitola 1.4.1), jelikož kresba tužkou mi v rámci této práce nepřišla jakožto nutná, vzhledem k implementaci pera a štětců. Ve výsledném projektu lze ve virtuální realitě kreslit na papír inkoustovým perem, štětcem kterému lze měnit barvu kresby za pomoci palety, ořezat papír, následně ho znovu slepit dohromady, kreslit perem rovné linie za pomoci vodítek, vložit ořezanou část papíru do modelu sloužícího jako předloha, papír lze do určité míry zvětšovat a zmenšovat a veškeré kresby, ať už perem nebo štětcem lze mazat. Celý projekt byl implementován v herním enginu Unity, virtuální zážitek je plynulý i na výkonostně slabším stroji a celý zážitek lze ovládat za pomoci interakcí v herním světě, bez využití nerealistických ovládacích prvků, s výjimkou posunutí uživatele před stůl po spuštění projektu za pomoci klávesnice.

### 6.2 Závěry z uživatelského testování

Mé poznatky z testování se shodují se závěry (viz podsekcce 5.1.2) vyplývajícími z grafů (obr. 5.1 až obr. 5.8). Interakce s papírem je implementována bez větších omezení a je možné například si papír vložit do modelu stolu, jelikož papír jako takový je čistě kinematický objekt, což není uživatelsky přívětivé. Uživatel může snadno provést něco, co by se v realitě nestalo, což vede k narušení zážitku. Jako úprava by bylo třeba vytvořit systém omezení pro objekt papíru, díky kterému by se jeho chování více přibližovalo fyzickému objektu, ale stále umožňovalo kreslení nástroji bez větších problémů.

Interakce s nástroji je další oblastí, na kterou by se v případném dalším vývoji bylo nutné zaměřit. Možnost rozložených nástrojů na pracovním stole, ze kterého si je uživatel může libovolně brát a používat je sice realistická, nicméně vede k potížím při výběru, pokud je jich vícero na jednom místě. To komplikuje detekci, který nástroj se uživatel snaží uchopit a dochází k úchopu nesprávných nástrojů.

Zároveň je systém držení nástroje za pomoci tisknutí předního tlačítka ovladače nešikovný pro mladší uživatele, pro které je ovladač příliš velký a pravidelně nedokáže manipulovat s ovladačem a držet stisknuté tlačítko, což vede k vypadávání nástroje z ruky. Způsob úchopu a držení nástrojů by vyžadoval dle mého názoru také další úpravy.

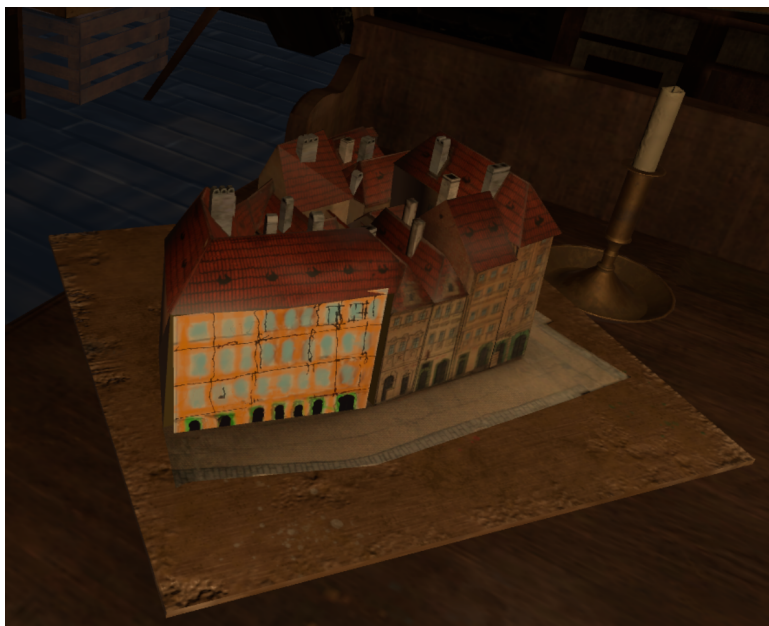
Pro uživatele by bylo přívětivější, pokud by se do zážitku implementoval systém návodů či tutoriálů k jednotlivým nástrojům, jelikož mezi různými nástroji jsou rozdíly

ve způsobu ovládání, které uživatel bez vysvětlení nemusí být schopen sám odhalit. pro většinu uživatelů nebyl zážitek dostatečně intuitivní, aby jim stačilo vysvětlení funkcionalit na začátku testování a bylo třeba se jim věnovat i během procesu testování.

Finálním poznatkem z testování je, že by bylo vhodné zvětšit plochu papíru, která reprezentuje výslednou stěnu. V tuto chvíli je velikost výsledné stěny vyznačená na papíru v jeho základní velikosti, nicméně tato plocha je na detailní práci příliš malá. Uživatel si sice může sám zvětšit papír a umožnit si tím pracovat na detailech, bylo by však uživatelsky přívětivější, kdyby mohl takto pracovat už od začátku - mnoho uživatelů možnost zvětšení papíru ani nevyužilo, přestože s ní byli obeznámeni.

### 6.3 Zhodnocení implementace autorem

Implementace požadavků v projektu rozhodně není bezchybná. Během testování se ukázalo, že množství ovládacích prvků je buď neintuitivních či se špatně ovládají, ale dohromady veškeré tyto prvky dávají ucelený zážitek, který v případě že je s uživatel seznámen s jeho omezeními, umožní vyzkoušet si na vlastní kůži tvorbu kusu Langweilova modelu Prahy ve virtuální realitě (obr. 6.1). Na základě zpětné vazby získané během testování, by určitě bylo možné zážitek dále vylepšit co se týče uživatelské přívětivosti a optimalizovat a poskytnout mi mnoho zkušeností, co se návrhu a implementace interakčních technik ve virtuální realitě týče.



**Obrázek 6.1.** Screenshot finálního výtvaru pořízený během uživatelského testování  
Všeobecně z mého pohledu tato implementace není bez vad a nedostatků, ale celkově vzato vyhovuje zadání práce.

## Zdroje

- [1] Lukáš Pospíšil [cit. 2022-05-19]. , Virtuální zážitek „Tvorba Langweilova modelu Prahy“. BP ČVUT FEL 2021.
- [2] JERALD, Jason. [cit. 2022-05-18]. The VR book: human-centered design for virtual reality. [San Rafael, CA]: Morgan & Claypool, [2016]. ACM book series. ISBN 978-1-97000-115-0.
- [3] Muzeum hlavního města Prahy - [online] [cit. 2022-05-19]. Dostupné na <http://www.langweil.cz/index.php>.
- [4] *Unity Technologies* [online] Manuál prostředí Unity, - [cit. 2022-01-14]. Dostupné na <https://docs.unity3d.com/Manual/index.html>.
- [5] *Unity Technologies* [online] Manuál prostředí Unity, metoda SetPixel - [cit. 2022-01-14]. Dostupné na <https://docs.unity3d.com/ScriptReference/Texture2D.SetPixel.html>.
- [6] *Unity Technologies* [online] Manuál prostředí Unity, Shader objekt - [cit. 2022-01-14]. Dostupné na <https://docs.unity3d.com/Manual/shader-objects.html>.
- [7] *Valve Corporation* [online] Manuál knihovny SteamVR, - [cit. 2022-01-14]. Dostupné na [https://valvesoftware.github.io/steamvr\\_unity\\_plugin/api/index.html](https://valvesoftware.github.io/steamvr_unity_plugin/api/index.html).
- [8] EMILIANA VT - [online] *GitHub projektu DokoDemoPainter*, - [cit. 2022-01-14]. Dostupné na <https://github.com/emilianavt/DokoDemoPainter>.
- [9] MIX AND JAM - [online] *GitHub projektu Splatoon's Ink System*, - [cit. 2022-01-14]. Dostupné na <https://github.com/mixandjam/Splatoon-Ink>.
- [10] David Arayan - [online] *GitHub projektu EzySlice*, - [cit. 2022-05-19]. Dostupné na <https://github.com/DavidArayan/ezy-slice>.
- [11] *Unity Technologies* [online] Manuál prostředí Unity, - [cit. 2022-05-18]. Dostupné na <https://docs.unity3d.com/ScriptReference/Camera.ScreenPointToRay.html>.
- [12] *Unity Technologies* [online] Manuál prostředí Unity, metoda RayCast - [cit. 2022-01-14]. Dostupné na <https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>.
- [13] *Unity Technologies* [online] Manuál prostředí Unity, metoda korutiny - [cit. 2022-01-14]. Dostupné na <https://docs.unity3d.com/Manual/Coroutines.html>.

- [14] *Unity Technologies* [online] Manuál prostředí Unity, komponent LineRenderer - [cit. 2022-01-14]. Dostupné na <https://docs.unity3d.com/Manual/class-LineRenderer.html>.

# Příloha A

## Dotazník vyplněný testovacími uživateli

### Testování tvorby Langweilova modelu ve VR

\*Povinné pole

1. Jakou verzi aplikace jsi používal/a? \*

*Označte jen jednu elipsu.*

VR1 - kovalmat

VR2 - papayrob

2. Kterou rukou pišeš? \*

*Označte jen jednu elipsu.*

Levou

Pravou

Levou i pravou stejně dobře.

3. Jak často používáš VR headset? \*

*Označte jen jednu elipsu.*

Nikdy jsem jej nepoužíval/a. *Přeskočte na otázku 6*

Velmi příležitostně jej používám (párkrát v roce). *Přeskočte na otázku 4*

Často (v průměru alespoň 1x měsíčně) *Přeskočte na otázku 4*

#### Zkušenosti s VR headsetem

4. Jaký způsob ovládání u headsetu jsi v minulosti používal/a?

*Zaškrtněte všechny platné možnosti.*

Volné ruce (bez ovladačů)

Standardní ovladače

Jiné (např. specializované ovladače)

5. Měl/a jsi někdy při práci s VR headsetem fyzické potíže (závrať, točení hlavy apod.)?

Označte jen jednu elipsu.

- Ano, výrazné.  
 Ano, ale jen mírné.  
 Ne, neměl jsem.

#### Tvorba grafického obsahu

6. Jaký je tvůj vztah ke kreslení na papír? \*

Označte jen jednu elipsu.

- Téměř vůbec si nekreslím, nemám na to talent.  
 Kreslím příležitostně  
 Kreslím často (v průměru alespoň 1x měsíčně)

7. Lepil/a jsi někdy papírové modely? \*

Označte jen jednu elipsu.

- Nikdy jsem papírový model nelepil/a.  
 Papírový model jsem několikrát slepil.  
 Tvořím často (v průměru alespoň 1x měsíčně)

8. Jaký je tvůj vztah ke tvorbě grafického obsahu (obrázky, 3D modely apod.) na počítači? \*

Označte jen jednu elipsu.

- Nikdy jsem grafický obsah netvořil/a. *Přeskočte na otázku 11*  
 Obsah tvořím jen příležitostně. *Přeskočte na otázku 9*  
 Tvořím často (v průměru alespoň 1x měsíčně) *Přeskočte na otázku 9*

#### Tvorba grafického obsahu detailně

9. Jaké editory pro tvorbu grafického obsahu používáš?

\_\_\_\_\_



10. Jaká zařízení při tvorbě používáš?

*Zaškrtněte všechny platné možnosti.*

- Počítačovou myš  
 Dotykový tablet bez pera  
 Dotykový tablet s perem  
 Jiné: \_\_\_\_\_

### Dotazník po testu

11. Jak se cítíš po fyzické stránce? Měl/a jsi nějaké potíže (závrať, točení hlavy apod.)?

\_\_\_\_\_

12. Jaký je tvůj nejvýraznější pocit z používání aplikace?

\_\_\_\_\_

13. Jak jsi spokojený s prostředím? \*

*Označte jen jednu elipsu.*

	1	2	3	4	5	
Velmi spokojený	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Velmi nespokojený

14. Jak jsi spokojený s kreslením perem? \*

*Označte jen jednu elipsu.*

	1	2	3	4	5	
Velmi spokojený	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Velmi nespokojený

15. Jak jsi spokojený s malováním štětcem? \*

*Označte jen jednu elipsu.*

	1	2	3	4	5	
Velmi spokojený	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Velmi nespokojený

16. Jak jsi spokojený s mazáním? \*

*Označte jen jednu elipsu.*

	1	2	3	4	5	
Velmi spokojený	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Velmi nespokojený

17. Jak jsi spokojený s řezáním? \*

*Označte jen jednu elipsu.*

	1	2	3	4	5	
Velmi spokojený	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Velmi nespokojený

18. Jak jsi spokojený se zvětšováním/zmenšováním papíru? \*

*Označte jen jednu elipsu.*

	1	2	3	4	5	
Velmi spokojený	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Velmi nespokojený

19. Jak jsi spokojený s přemístováním papíru? \*

*Označte jen jednu elipsu.*

	1	2	3	4	5	
Velmi spokojený	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Velmi nespokojený

20. Jak jsi spokojený/á s korespondencí pozice virtuálního nástroje a reálného ovladače? \*

*Označte jen jednu elipsu.*

	1	2	3	4	5	
Velmi spokojený	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Velmi nespokojený

21. Co se ti při práci v aplikaci nejvíc líbilo?

\_\_\_\_\_

22. Co se ti při práci v aplikaci nejvíc nelíbilo?

\_\_\_\_\_

23. Je něco, co v aplikaci chybí?

\_\_\_\_\_

24. Jaký je tvůj názor na rozmanitost nástrojů? \*

*Označte jen jednu elipsu.*

	1	2	3	4	5	
Nástrojů je málo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Nástrojů je moc.

25. Jaký je tvůj názor na vzdálenost aktivace nástrojů (např. kdy začne pero kreslit)? \*

*Označte jen jednu elipsu.*

- Nechal/a bych tak jak to je teď.
- Preferoval/a bych kratší vzdálenost (nástroje mohou vynechávat).
- Preferoval/a bych delší vzdálenost (nástroje mají tendenci aktivovat se když to není žádáno).
- Preferoval/a bych možnost aktivovat nástroj pomocí stisknutí tlačítka.
- Jiné: \_\_\_\_\_

26. Přišla ti práce v aplikaci podobná kreslení na papír \*

*Označte jen jednu elipsu.*

	1	2	3	4	5	
Velmi podobná	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Vůbec nebyla podobná

---

Obsah není vytvořen ani schválen Googlem.

Google Formuláře

# Příloha B

## Readme soubor

BP Matyáš Kovař

Kreslení na papír ve VR

Projekt předpokládá vlastnictví virtuálně-realistního headset a ovladačů s ním spárovaných.

## Návod na spuštění sestavené verze:

Nainstalujte si, prosím, software na propojení Vašeho VR headsetu s počítačem (například [Windows Mixed Reality Portal](<https://www.microsoft.com/en-us/p/mixed-reality-portal/9ng1h8b3zc7m#activetab=pivot:overviewtab>) pro Windows mixed VR headset nebo [Oculus](<https://store.facebook.com/quest/setup/> pro headsety Oculus)).

Dále si nainstalujte Steam a rozšíření [SteamVR] (<https://store.steampowered.com/app/250820/SteamVR/>).

Zapněte obě aplikace, připojte headset a ovladač.

Stáhněte přílohu obsahující sestavenou verzi BP a otevřete ji.

Dále spusťte program BPkovalmat.exe, čímž se spustí sestavená verze virtuálního zážitku.

## Ovládání aplikace

V sestaveném a spuštěném projektu se uživatel může poposunout v rámci scény za pomoci kláves na klávesnici.

w - přisunout ke stolu

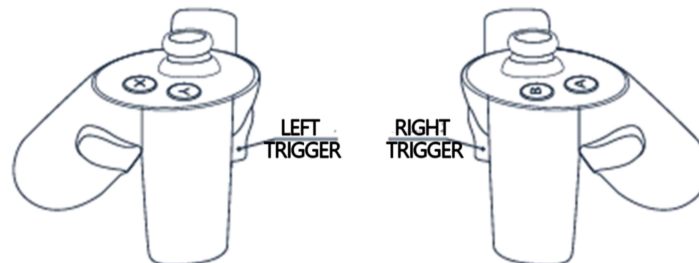
s - odsunout od stolu

a - posunout doleva vzhledem ke stolu

d - posunout doprava vzhledem ke stolu

q - přiblížit se k podlaze

e - oddálit se od podlahy



Všechny ostatní interakce již probíhají za pomoci virtuálně realitních ovladačů:

Pomocí tlačítka GrabPinch (přední trigger ovladače) lze uchopit nástroje do ruky, která je ovládána ovladačem, na kterém došlo ke stisknutí tlačítka.

Aby uživatel mohl nástroj zvednout, je k němu nejdříve třeba ruku přiblížit a ve chvíli, kdy se předmět zvýrazní zelenou barvou, tak předmět lze stisknutím tlačítka GrabPinch uchopit do ruky.

Uživatel nástroj drží tak dlouho, dokud drží tlačítko grabPinch stisknuté. Ve chvíli, kdy tlačítko uvolní, nástroj bude upuštěn.

K dispozici jsou následující nástroje:

inkoustové pero (po pravé straně papíru): Toto pero lze využít ke kreslení. Lze s ním kreslit volně (přiblížení špičky pera k papíru, dokud nezačne kreslit) nebo za pomoci vodítek lze kreslit rovné čáry.

Na kreslení za pomoci vodítek se dá přepnout tím, že se perem jednou dotknete objektu pravítka (černý trojúhelník napravo od knihy na stole). Tím se pravítko zvýrazní zelenou barvou, což indikuje,

že došlo k přepnutí. Dále při přiblížení pera k papíru se již nebude kreslit, ale pouze dojde k vybrání počátečního bodu rovné čáry a začne se červenou barvou

vykreslovat ukazatel, značící koncový bod čáry. Při zvednutí pera z papíru dojde k nakreslení vyznačené čáry. Přepnout na volnou kresbu lze opět dotknutím se pravítka perem.

malý a velký štětec (po pravé straně papíru u palety): Štětce lze využít ke kreslení. Nicméně nejdříve je třeba vybrat barvu jakou budou kreslit - štětec po spuštění scény barvu vybranou nemá.

To se provede dotknutím se štětcem palety na místě, kde je na paletě barva, kterou chceme zvolit. Výběr barvy lze vizuálně potvrdit tím, že bude špička

štětce mít stejnou barvu, jako ta, kterou jsme zvolili. Štětcem lze kreslit pouze volně, ale má přidanou mechaniku, že čím blíže je špička štětce k papíru,

tím tlustší je kresba (simulace „tlačení“ na štětec).

Štětci lze provádět i mazání kresby štětcem. Na to je potřeba mít štětec bez vybrané barvy - ten pak místo kresby barvu rozředí a eventuálně maže.

Štětec bez barvy se získá otřením obarveného štětce o pomalovaný hadřík (obdélník nad papírem na stole).

Malý a velký štětec jsou funkčně identické, liší se pouze v průměru kresby.

mazátko (po levé straně papíru): Jedná se o modro-bílé mazátko, který lze mazat kresbu perem.

Funkčně je identické peru, jen s rozdílem, že kresbu maže a

má větší průměr kresby.

perořízek (v horní části stole, nalevo od kalamáře): Perořízek slouží k rozřezání papíru. Funkcí je podobný kreslení s vodítky. Při přiblížení perořízku

k papíru dojde k vybrání počátečního bodu řezu a následně se začne vykreslovat kudy povede řez papírem. Je nutné vybrat druhý bod, kterým řez povede dostatečně

daleko od bodu počátečního (mechanika pro možnost zrušení řezu, který jsme započali na špatném místě). Ve chvíli kdy je druhý bod řezu dostatečně daleko,

dojde ke změně barvy vykresleného ukazatele řezu z červené na modrou. Ve chvíli kdy je druhý bod řezu dostatečně daleko se při zvednutí perořízku od

papíru provede řez. Řez bude veden celým papírem skrze počáteční a koncový bod, které jsme vybrali perořízkem.

paleta (po pravé straně papíru): Jak bylo zmíněno v sekci ovládní štětců, paleta je statická a slouží pouze k vybrání barvy štětcem (pozor v kalíšku na konci palety je černá barva,

ne voda - držel jsem e při výběru barev textury palety)

pravítko (po levé straně knihy): Jak bylo zmíněno v sekci ovládní pera, pravítko je pouze statické a slouží k přepínání mezi kresbou perem s vodítky a bez.

hadřík- pokreslený (nad papírem): Jak bylo zmíněno v sekci ovládní štětců, hadřík je statický a slouží pouze ke smytí barvy ze štětce.

papír: Papír slouží ke kreslení, jedná se o jediný objekt ve scéně, na který lze kreslit. Papír lze přesouvat tím, že ho uživatel uchopí jednou rukou,

nicméně doporučuji ho držet co nejvíce vodorovný (v rovině stolu). Papír nemá fyziku, jedná se o kinematický objekt. Dále lze papír zvětšovat a zmenšovat. To se provede uchopením papíru do obou rukou

a následným pohybem rukou k sobě či od sebe. Papír bude měnit velikost dle poměru vzdáleností rukou při začátku úchopu a v průběhu.

Rozřezané kusy papíru lze také přesouvat či jím měnit velikost.

Na papíru je vyznačená velikost kresby, která odpovídá chybějící stěně modelu.

model domu (po levé straně papíru): S modelem lze otáčet uchopením jeho podstavu a následným táhnutím za pomoci ruky, kterou jsme podstavu uchopili.

Dále lze do modelu vložit buď papír celý nebo jeho uříznutý kus (vlození proběhne, pokud se papír vloží do chybějící stěny domu).

Doporučuji před vkládáním odříznutých kusů papíru papír nejdříve zvětšit a vkládat ho vodorovně.

kniha (pravý horní roh stolu): Kniha obsahuje základní popis scény, příklad vzhledu stěny modelu a dva zarámované nápisy na druhé straně.

„Vrátit nástroje“ a „Slepit papír“. Tyto nápisy slouží jako tlačítka, která lze stisknout buď rukou nebo objekty ve scéně.

„Vrátit nástroje“ vrátí veškeré nástroje na místo, na kterém byli po spuštění scény.

„Slepit papír“ znovu spojí papír po posledním řezu.

##Návod na spuštění Unity adresáře

Po stažení repositáře otevřete složku BPkvalmat za pomoci Unity Hub nebo přímo Unity.

Tato složka obsahuje celý projekt.

Pro otevření doporučuji použít verzi Unity, které tento projekt doporučuje, neboť v ní byl vytvořen (jedná se o verzi 2020.3.21f1).


Po otevření tohoto projektu otevřete v prohlížeči v dolní části Unity rozhraní složku Assets>Scenes a poklikejte na scénu jménem "Puda"

Toto je testovací scéna. Po kliknutí na ikonu trojúhelníku v horní části Unity rozhraní by se mělo zapnout SteamVR a do VR brýlí by se měla začít promítat scéna.

Pokud nedojde k zapnutí SteamVR, je třeba jít do záložky Edit>Project Settings>XR Plug-In Management, odkliknout OpenVR Loader, zapnout scénu,

vypnout jí a opět tuto možnost zaškrtnout. Poté by se již projekt měl zapnout v pořádku.





Ovládání je identické se sestavenou verzí.

Veškeré skripty tohoto projektu jsou v adresáři Assets>Scripts a Assets>EzySlice.