



## Zadání diplomové práce

<b>Název:</b>	Ovládání dronu za pomoci technik hand-trackingu
<b>Student:</b>	Bc. Matyáš Sojka
<b>Vedoucí:</b>	Ing. Tomáš Nováček
<b>Studijní program:</b>	Informatika
<b>Obor / specializace:</b>	Softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	do konce letního semestru 2023/2024

### Pokyny pro vypracování

S nástupem nových technologií na ovládání uživatelského rozhraní přichází nové možnosti, ale i nové výzvy. Sledování rukou uživatele s pomocí kamer je čím dál oblíbenější a přesnější, a tak jej můžeme používat na složitější úkony, než pouhé otevírání dveří. Jedním z možných využití je například ovládání drona, ať fyzického či virtuálního.

- 1) Analyzujte možnosti sledování rukou uživatele za pomoci kamer. Zaměřte se převážně na senzor OAK-D-Lite.
- 2) Proveďte rešerši systémů, které využívají sledování rukou uživatele jako hlavní způsob ovládání.
- 3) Naimplementujte ovládání virtuálního drona za pomoci technik popsaných v analytické části. Zaměřte se na přesnost a robustnost ovládání.
- 4) Svoje výstupy ověřte za pomoci jednoduché aplikace.



Diplomová práce

# OVLÁDÁNÍ DRONU ZA POMOCÍ TECHNIK HAND-TRACKINGU

**Bc. Matyáš Sojka**

Fakulta informačních technologií  
Katedra softwarového inženýrství  
Vedoucí: Ing. Tomáš Nováček  
11. ledna 2024

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2024 Bc. Matyáš Sojka. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.*

Odkaz na tuto práci: Sojka Matyáš. *Ovládání dronu za pomoci technik hand-trackingu*. Diplomová práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

# Obsah

Poděkování	vi
Prohlášení	vii
Abstrakt	viii
Seznam zkratek	ix
Úvod	1
<b>1 Analýza</b>	<b>3</b>
1.1 Analýza současného stavu	3
1.1.1 Možnosti sledování rukou uživatele	3
1.1.2 Systémy využívající hand-tracking	4
1.1.3 Existující řešení	4
1.2 Analýza HW	5
1.2.1 Drony	5
1.2.2 Leap Motion Controller	6
1.2.3 OAK-D-Lite	6
1.3 Analýza UI	7
1.4 Vytvoření aplikace s podporou hand-trackingu	8
1.4.1 OpenCV & MediaPipe	8
1.4.2 Unity OpenXR & Ultraleap Plugin	9
1.4.3 DepthAI & OAK for Unity	10
<b>2 Návrh</b>	<b>11</b>
2.1 Technologie	11
2.2 Ovládání	12
2.3 Scéna	14
2.4 Návrh UI	17
<b>3 Implementace</b>	<b>21</b>
3.1 Předání a zpracování dat	21
3.2 Ovládání	22
3.2.1 Pohyb dopředu a dozadu	22
3.2.2 Pohyb do stran a rotace	23
3.2.3 Pohyb nahoru a dolů	23
3.2.4 Ovládání klávesnic	24
3.3 Gamifikace	24
3.4 UI aplikace	25
3.5 Problémy a omezení	27
3.6 Možnosti dalšího rozšíření	28

<b>4 Testování</b>	<b>29</b>
4.1 Průběh testování . . . . .	29
4.2 Výsledky testování . . . . .	30
4.2.1 Ovládání . . . . .	30
4.2.2 UI a celkové hodnocení . . . . .	32
4.3 Reakce na testování . . . . .	34
<b>Závěr</b>	<b>37</b>
<b>A Slovník pojmů</b>	<b>39</b>
<b>B Spuštění aplikace</b>	<b>41</b>
<b>C Kompletní výsledky testování</b>	<b>43</b>
<b>Obsah přiloženého média</b>	<b>55</b>

## Seznam obrázků

1.1	Hand-tracking pomocí senzoru Leap Motion Controller [22] . . . . .	6
1.2	Senzor OAK-D-Lite [25] . . . . .	7
1.3	MediaPipe model dlaně s prsty mapovaný na dlaň uživatele [32] . . . . .	9
2.1	Zapojení senzoru Ultraleap . . . . .	11
2.2	Souřadnicový systém senzoru Ultraleap [41] . . . . .	12
2.3	Proces snímání rukou uživatele za běhu aplikace . . . . .	13
2.4	Návrh prázdné tréninkové scény (vlevo) a jednoduché scény s několika překážkami a checkpointy (vpravo) . . . . .	14
2.5	Návrh gamifikace aplikace . . . . .	16
2.6	Návrh dynamického pohybu kamery . . . . .	18
2.7	Návrh UI se sidebarem s informacemi o letu . . . . .	19
2.8	Návrh UI ve fullscreen modu . . . . .	19
2.9	Návrh jednoduchého menu . . . . .	20
3.1	Výuková scéna s několika překážkami a popsáním ovládním . . . . .	25
3.2	Uživatelské rozhraní aplikace při letu dronem . . . . .	26
3.3	Uživatelské rozhraní aplikace v menu . . . . .	27
4.1	Jeden z testerů při průletu výukovou scénou . . . . .	30
4.2	Výsledky hodnocení ovládním hand-trackingem . . . . .	31
4.3	Výsledky hodnocení ovládním klávesnicí . . . . .	31
4.4	Výsledky hodnocení uživatelského rozhraní . . . . .	33
4.5	Aktualizované UI menu . . . . .	35

## Seznam tabulek

2.1	Návrh ovládním pomocí hand-trackingu . . . . .	15
4.1	Výsledky srovnání ovládním hand-trackingem a klávesnicí . . . . .	32
4.2	Výsledky srovnání užitečnosti prvků UI . . . . .	33

*Rád bych poděkoval především vedoucímu mé práce, Ing. Tomáši Nováčkovi, za jeho cenné a věcné rady, vlídný přístup a vedení v průběhu tvorby této práce. Poděkování patří i všem dobrovolníkům, kteří se zúčastnili uživatelského testování. V neposlední řadě bych chtěl poděkovat přátelům a rodině za ochotu a neustálou podporu.*



## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 11. ledna 2024

## Abstrakt

V dnešní době přibývá zajímavých možností, jak mohou uživatelé interagovat s virtuálním světem. Jedním ze způsobů, který je stále využívanější, je hand-tracking – snímání rukou uživatele.

Tato práce se zabývá návrhem a implementací jednoduché aplikace, která umožňuje ovládání virtuálního dronu právě pomocí hand-trackingu. Využívá k tomu senzor Ultraleap Stereo IR 170. Uživatel si může ovládání vyzkoušet na několika vytvořených scénách, kde je jeho cílem nasbírat checkpointy v co nejkratším čase. Vedle toho může využít i jednoduché ovládání klávesnicí.

**Klíčová slova** hand-tracking, Ultraleap, dron, rozpoznávání gest, zpracování obrazu, Unity

## Abstract

Nowadays, there are more and more interesting ways for users to interact with the virtual world. One of the ways that is increasingly being used is hand-tracking – sensing the user's hands.

This thesis deals with the design and implementation of a simple application that allows controlling a virtual drone by hand-tracking. It uses the Ultraleap Stereo IR 170 sensor to do so. The user can test the hand tracking controls in several virtual scenes, where the goal is to collect checkpoints in the shortest time possible. In addition to this, he can also use simple keyboard controls.

**Keywords** hand-tracking, Ultraleap, drone, gesture recognition, image processing, Unity

## Seznam zkratek

API	Application Programming Interface
AR	Augmented Reality
BW	Black and White
CPU	Central Processing Unit
FPS	Frames Per Second
FPV	First-Person View
GPU	Graphics Processing Unit
HMD	Head Mounted Display
MR	Mixed Reality
RAM	Random Access Memory
RGB	Red, Green, Blue
SOTA	State of the Art
TLD	Tracking-Learning-Detection
ToF	Time of Flight
VR	Virtual Reality
XR	Extended Reality



# Úvod

V dnešní době se technologie stále více zapojují do našeho každodenního života a přináší nové způsoby interakce s virtuálním světem. Bezdotykové uživatelské rozhraní tak již není hudbou budoucnosti, ale již si pomalu hledá – a v některých případech si již našlo – cestu nejen do našich domovů.

Na oblíbenost stále více získává tzv. hand-tracking, při kterém je snímán pohyb a rotace rukou uživatele. Hand-tracking je dnes již na takové úrovni, že je například běžnou součástí VR headsetů, či se začíná používat i v palubních deskách aut. V této práci se jsem se zabýval konkrétně optickým markerless trackováním.

S rostoucí přesností snímání rukou uživatele se nabízí využít jeho potenciál i v odvětvích, ve kterých by nás to ještě před pár lety nenapadlo. Lze ho využít například k ovládání dronu? Právě na tuto otázku přináší tato diplomová práce odpověď.

## Cíle a motivace práce

Konkrétním výstupem práce bude jednoduchá aplikace, která bude implementovat ovládání virtuálního dronu pomocí optického hand-trackingu. Právě implementace a optimalizace tohoto ovládání budou hlavním cílem této práce. Důraz bude přitom kladen na robustnost a přesnost ovládání.

Práce rozhodně nebude určena jen pro uživatele, kteří mají předchozí zkušenost s ovládáním reálného dronu. Ovládání bude navrženo tak, aby si ho i uživatelé s nulovou zkušeností mohli během chvíle osvojit. Cílem bude, aby bylo ovládání snadno pochopitelné, imerzivní a přitom přesné.

Téma jsem si zvolil, neboť technika hand-trackingu je velice aktuální, zajímavá a čím dál více využívaná. Hlavním přínosem mé práce má být alternativní, neotřelý, ale stále dostatečně intuitivní způsob ovládání virtuálního dronu. Může přispět k tomu, aby se v budoucnu hand-tracking využil i pro ovládání reálného dronu.

## Struktura práce

Práce má následující strukturu: analýza, návrh, implementace a testování. Jednotlivé kapitoly jsou tedy věnovány dílčím fázím vývoje. Jejich výsledky jsou pak shrnuty v závěru.

V rámci analýzy jsem se zabýval existujícími systémy využívající hand-tracking. Dále jsem analyzoval hardware dronů s důrazem na jejich ovládání a blíže jsem se seznámil se senzory OAK-D-Lite a Leap Motion Controller (Ultraleap). Následovalo prověření uživatelského rozhraní aplikací, které využívají hand-tracking a gesture recognition. Nakonec jsem prozkoumal úskalí vytvoření aplikace podporující hand-tracking.

Na základě této analýzy jsem navrhl aplikaci pro účely ovládání virtuálního dronu za pomoci hand-trackingu. Návrh počítá s využitím Ultraleap Controlleru pro samotný hand-tracking, pro vývoj se pak hodí nástroj Unity a rozšíření Ultraleap Plugin for Unity. Dále jsem se v návrhu zabýval ovládáním pohybem a rotací ruky, následně scénou, ve které se bude dron pohybovat a konečně uživatelským rozhraním aplikace.

Implementace probíhala ve frameworku Unity, o hand-tracking se postaral senzor Ultraleap Stereo IR 170. Hlavní důraz byl přitom kladen právě na ovládání hand-trackingem. V rámci práce jsem vytvořil aplikaci ve formě jednoduché hry, ve které je uživatelským cílem proletět checkpointy v co nejkratším čase. Dále jsou v této kapitole popsány problémy, na které jsem při implementaci narazil a možnosti budoucího rozšíření.

Výsledná aplikace byla nakonec podrobena uživatelskému testování. To pomohlo odpovědět na otázku, zda je ovládání intuitivní a vhodné pro létání s dronem. Nakonec jsem adresoval připomínky, které vzešly z dotazníku a opravil některé chyby, které při testování vyšly najevo.

# Kapitola 1

## Analýza

*V rámci analýzy jsem se zabýval existujícími systémy využívající hand-tracking. Dále jsem analyzoval HW dronů s důrazem na jejich ovládání a blíže jsem se seznámil se senzory OAK-D-Lite a Leap Motion Controller. Následovalo prověření uživatelského rozhraní aplikací, které využívají hand-tracking a gesture recognition. Nakonec jsem prozkoumal úskalí vytvoření aplikace podporující hand-tracking.*

### 1.1 Analýza současného stavu

*Analýza současného stavu přinesla vzhled do problematiky hand-trackingu. Dále rozebrala jednotlivé možnosti a technologie a nakonec přiblížila systémy, které tyto funkce využívají. Nakonec poskytla přehled existujících řešení.*

#### 1.1.1 Možnosti sledování rukou uživatele

*V rámci této kapitoly jsem se podrobněji seznámil s možnostmi sledování rukou uživatele. Zaměřil jsem se především na markerless trackování, zmiňuji zde však i alternativy.*

Snímání těla uživatele a ovládání uživatelského rozhraní vlastním tělem je stále přesnější a používanější. Snaha oprostít uživatele od ovladačů, klávesnic, joysticků a podobných pomůcek má typicky za cíl větší imerzivitu a přirozenost.

Obecně existují dva způsoby [1], jak lze snímat tělo uživatele – aktivní a pasivní. První způsob vyžaduje určité rozhraní (např. ovladač), které buď přijímá nebo vysílá signál o pozici ruky. Tuto technologii využívá například první verze HTC Vive [2]. Pasivní trackování vyžaduje umístění reflexních bodů, které jsou sledovány kamerou – typické pro motion capture. Dalším typem je markerless trackování, které nevyžaduje žádné další rozhraní ani reflexní body. Senzor sleduje přímo uživatele a jeho pohyb a gesta.

Jedním z prvních masově rozšířených markerless zařízení byl Kinect, který využívá hloubkové kamery pro rozpoznání celého těla uživatele [3]. První verze z roku 2010 pracovala na technologii structured light – dvě infračervené kamery snímají scénu, dopočítávají hloubku a tím i pozici uživatele. Druhá generace Kinectu pak pracovala s technologií ToF (Time of Flight) [4] – do scény jsou vyslány světelné paprsky, hloubka se získá na základě změny amplitudy odraženého světla.

Na bázi kamer a infračervených LED diod funguje i senzor Leap Motion Controller [5]. Soustředí se téměř výhradně na trackování rukou uživatele. Nabízí vlastní trackovací software, který dokáže rozpoznat jednotlivé články prstů i klouby. Rovněž je možné propojit a synchronizovat několik těchto senzorů a vytvořit tak ještě robustnější řešení [6]. Narozdíl od Kinectu se

hodí spíše na kratší vzdálenosti (řádově desítky centimetrů).

Když na to však přijde, je možné sledovat pohyb uživatele i například pomocí obyčejné webkamery a výstup zpracovávat kupříkladu pomocí knihovny OpenCV. Tento koncept si vzali k srdci tvůrci senzoru OAK-D-Lite. Ten používá dvě monokulární kamery pro stereoskopické vnímání hloubky a jednu RGB 4K kameru pro samotné zachycení barevného obrazu. Spolu s tímto optickým aparátem obsahuje výkonné procesory, které zpracovávají data pro detekci hran, rozpoznání objektů a tváří, trackování objektů apod [7]. Blíže je tento senzor popsán v kapitole 1.2.

Pro úplnost bych rád zmínil ještě haptické obleky. Jedná se o další způsob určení polohy a pózy uživateleova těla. Tento způsob je přesnější a nabízí navíc haptický feedback, je však obecně dražší a méně příjemný a vyžaduje specializovaný hardware, který si uživatel nasadí. Pro účely této práce jsem se proto rozhodl využít raději optické markerless trackování.

### 1.1.2 Systémy využívající hand-tracking

*V této kapitole jsem identifikoval systémy, které umožňují uživateli ovládnání skrze optický markerless hand-tracking. Zmíněné systémy spadají do kategorie extended reality (XR).*

Jak již bylo zmíněno, markerless hand-tracking může přinést větší imerzivitě a přirozenější ovládnání. Není tedy divu, že systémů, které tuto funkci podporují, přibývá – především pak v odvětví VR, AR a MR technologií.

Firma Oculus, respektive Meta, se rozhodla obohatit hand-trackingem svůj VR headset Quest 2. Využívají inside-out trackování pomocí senzorů umístěných na HMD. Systém v reálném čase rozpozná uživateleovy dlaně, na které namapuje 3D model a zobrazí je uživateli ve VR. Momentálně systém podporuje rozpoznání tří gest, a sice „Point and Pinch“, „Point and Drag“ a „Palm Pinch“ [8]. Obdobný princip i funkce pak využívá i headset Vive Focus 3 od společnosti HTC [9].

Headset Pimax nepodporuje hand-tracking přímo, ale jeho tvůrci ve spolupráci s firmou Ultraleap vytvořili montovatelný modul [10], který tuto funkci přidá. Dříve zmíněný Leap Motion Controller pak lze použít i se staršími VR headsety od HTC či Oculus a rozšířit je tak o sledování rukou uživatele [11].

Další oblastí, ve které má hand-tracking své uplatnění, je rozšířená a smíšená realita. Zde můžeme zmínit například chytré MR brýle Hololens 2. Ty díky hand-trackingu umožňují prostou interakci s virtuálními objekty. Není již potřeba vykreslovat 3D model virtuálních rukou, jelikož uživatel vidí skrze brýle své vlastní ruce. Podporu sledování rukou mají dále kupříkladu AR brýle Nreal Light.

### 1.1.3 Existující řešení

*Již vzniklo několik prací, které se zabývají alternativním ovládnáním dronů pomocí zpracování obrazu. Existují dokonce komerční řešení, které implementují rozpoznávání gest uživatele v rámci ovládnání dronů. V této kapitole zmiňuji několik z nich.*

Objevil jsem několik existujících řešení, které se zabývají podobnou tematikou jako tato práce. V rámci implementace softwaru FollowMe popisují autoři [12], jak vytvořili autonomního drona, který sleduje uživatele a následuje jeho pohyb. Využili přitom principu tracking-learning-detection (TLD) [13] – výstup z kamery je analyzován detektorem, který na základě strojového učení opravuje chyby při detekci uživatele. Po rozpoznání postavy uživatele se ji dron snaží udržet ve středu svého pohledu, přičemž si drží konstantní vzdálenost.

Zajímavý přístup zvolili na univerzitě Politecnico di Torino [14]. Navržený framework umožňuje ovládnání kvadrokoptéry pomocí celého těla. Jako snímač využili Microsoft Kinect, který je k tomuto účelu vhodný. Horizontální pohyb dronu je řízen náklonem horní poloviny těla, rotace



a vertikální pohyb pak pohyby rukou. Aby zvýšili robustnost systému, signál byl dronu vyslán pouze tehdy, když byl pohyb uživatele dostatečně znatelný (větší, než určitý práh).

Autoři videa Hand Gesture Drone Control [15] implementovali pomocí knihovny OpenCV jednoduchý gesture recognition systém, kterým lze pohybovat dronem ve čtyřech směrech (nahoru, dolů, doleva, doprava) pomocí dlaně. Nevyužívají žádný externí senzor, ale samotnou kameru dronu. Zároveň ošetřili systém tak, aby bylo dron možné ovládat pouze pokud kamera vidí obličej uživatele, který se tak musí pohybovat spolu s dronem. Rotaci dronu tato práce neřeší.

V článku Hand-worn Haptic Interface for Drone Teleoperation [16] popisují autoři, jak vytvořili systém pro ovládání dronu s využitím motion capture. Snímaná rukavice osazená reflexními body navíc poskytovala uživateli haptický feedback, pokud se dron blížil k překážce – jednalo se tedy o jednoduchý antikolizní systém. Samotné ovládání fungovalo na následujícím principu: uživatel začal s rukou v určité výchozí pozici a směr pohybu dronu byl pak řízen relativním pohybem ruky z výchozí pozice. Rozsah pohybu byl vždy vynásoben určitým škálovacím faktorem, pohyb ruky o několik centimetrů tak mohl znamenat pohyb dronu až o několik metrů. Autoři opět zanedbali rotaci dronu a soustředili se pouze na relativní pohyb ve všech 3 osách. Výstup byl ověřen jak na fyzickém, tak na virtuálním dronu - pomocí jednoduché aplikace v Unity.

Již zmíněný Leap Motion Controller využili autoři článku Web-based interactive drone control using hand gesture [17]. Pohyb dronu je opět navázán na vychýlení dlaně z iniciální pozice, je zde však bráno v potaz i natočení dronu, které je vázáno na rotaci dlaně kolem svislé osy. Podobně jako v článku, kde využili Kinect [14], i zde je ovládání regulováno pomocí thresholdingu. Tento threshold pro jednotlivé směry pohybu pak autoři našli experimentálně. Výstupem byla webová aplikace, která umožňovala přepínat mezi ovládáním tlačítka a hand-trackingem. Navíc zaznamenávala trajektorii letu dronu, která mohla být podrobena další analýze.

Existují rovněž komerční drony, které dokážou rozpoznat gesta uživatele a dovoluují tak jednoduché ovládání bez jakéhokoliv ovladače. Například dron DJI Spark dokáže navíc kopírovat pohyb uživatelské ruky, snímat a následovat pohyb uživatele v prostoru a reagovat na jednoduchá gesta [18]. Sledování uživatele v prostoru nabízí i další drony – samozřejmostí je u nich real-time zpracování obrazu a detekce objektů, což může navíc sloužit i k zabránění kolize při letu.

Unikátní přístup k ovládání pak zvolili tvůrci Firefly Drone [19]. Vytvořili ovladač ve formě náramku, který má v sobě zabudovaný gyroskop. Let dronu se pak ovládá nakláněním dlaně. Pohyb dronu je tak vždy relativní vůči uživateli, nikoliv vůči dronu samotnému – dron může být libovolně natočen a pohyb vykoná vždy ve směru náklonu uživatelské ruky.

## 1.2 Analýza HW

*V analýze hardwaru jsem se zabýval obecně drony jako takovými, s důrazem na jejich fungování a ovládání. Typické je ovládání pomocí ovladače s dvěma joysticky, avšak existují i alternativy, které využívají zpracování obrazu. Dále jsem se zaměřil na senzor OAK-D-Lite a Leap Motion Controller.*

### 1.2.1 Drony

Příchod a masové rozšíření dronů ulehčilo život filmařům, přineslo nový druh zábavy pro fanoušky vzdušných závodů nebo nabídlo další způsob doručení balíčků. Drony mají různé tvary a velikosti, od kapesních hraček až po metrové stroje. Jak vlastně ale takový dron funguje?

Nejčastějším druhem je tzv. kvadrokoptéra, dron se čtyřmi vrtulemi. Vrtule jsou umístěny na rohy pomyslného čtverce. Sousední vrtule se otáčejí opačným směrem, protilehlé vrtule pak točí stejným směrem. Pokud všechny vrtule rotují stejnou rychlostí, je dron stabilní. Pro pohyb dopředu je třeba přední vrtule zpomalit a naopak ty zadní zrychlit – obdobně pro pohyb v dalších směrech. Tuto synchronizaci má na starost řídicí jednotka, která reaguje na pokyny z vysílačky



■ **Obrázek 1.1** Hand-tracking pomocí senzoru Leap Motion Controller [22]

operátora.

Operátor ovládá dron nejčastěji pomocí ovladače s dvěma joysticky. Levý joystick má typicky na starost kontrolu výšky letu a rotaci dronu, pravý pak pohyb v horizontální rovině. Dron bývá vybaven kamerou, která napomáhá uživateli k lepší orientaci při letu, a zároveň umožňuje ovládat dron i na velké vzdálenosti. Výstup z kamery je přesměrován do obrazovky, nebo do virtuálních brýlí. Existují však i alternativní způsoby ovládání, které jsem již zmínil v kapitole 1.1.3.

Fakt, že ovládání dronu bez ovladače je aktuální a má smysl se jím zabývat, potvrzuje i práce [20]. V ní nasimulovali autonomní chování dronu, který reagoval na gesta a pokyny zúčastněných uživatelů. Téměř polovina účastníků se pak shodla, že ovládání pomocí gest a hlasu je pro ně přirozenější a intuitivnější.

## 1.2.2 Leap Motion Controller

Tento ovladač od firmy Ultraleap je v současné době tím nejrozšířenějším hardwarem pro hand-tracking na světě. O samotné snímání rukou se starají dvě monochromatické, infračervené kamery s rozlišením 640 x 240 pixelů a snímkovou frekvencí 120 FPS [5]. Na rozdíl od zařízení jako například Kinect pracuje lépe na menší vzdálenosti, ideálně od 10 cm do 60 cm, maximálně pak do 1 m.

Co však činí tento senzor jedinečným, je jeho přesnost a odezva. Je totiž citlivý na pohyb už od 0,7 mm s odezvou 26 ms. Dokáže zaznamenat pohyb celé dlaně a prstů, ale i jednotlivých kostí a kloubů, a to dokonce i když dojde k jejich okluzi. Široké zorné pole kamer (140° x 120°) umožňuje práci se senzorem umístěným na stole, nebo ho lze připevnit např. na VR brýle. Leap Motion Controller lze dále rozšířit o podložku Stratos, která pomocí ultrazvukových vln poskytuje haptický feedback [21].

Při vývoji s využitím ovladače Leap Motion lze využít API LeapC, které je primárně v jazyce C, ale existují i adaptace do dalších jazyků (Python, C#, ...). Dále jsou k dispozici rozšíření pro Unity a Unreal Engine, které umožňují použít ovladač spolu s těmito herními enginy.

## 1.2.3 OAK-D-Lite

Senzor OAK-D-Lite je, slovy výrobců, považován za „Švýcarský nůž počítačového vidění“ [23] a vzhledem k jeho výkonu, univerzálnímu použití a kompaktním rozměrům jim nelze než dát za



■ **Obrázek 1.2** Senzor OAK-D-Lite [25]

pravdu. Obsahuje pár BW stereo kamer (480p, 120 FPS) pro detekci hloubky a jednu RGB kameru (4K, 35 FPS) pro samotný barevný video výstup.

OAK-D-Lite dostupný ve dvou variantách: Auto-Focus (AF) a Fixed-Focus (FF) [24]. AF zvládá lépe ostření na menší vzdálenost (méně než 30 cm), ale má problémy s vibracemi a nehodí se tak pro ukotvení na pohybující se vozidla, drony apod. Naopak verze FF nezvládne plně zaostřit na vzdálenost pod 30 cm, ale poradí si lépe se silnějšími vibracemi. Nutno říct, že tyto parametry se týkají pouze RGB kamery. Stereo grayscale kamery jsou u všech OAK senzorů vždy Fixed-Focus, ale zvládají ostřit už na 10 cm vzdálené objekty. Zda bude pro potřeby této práce vhodnější varianta AF nebo FF bude záležet na návrhu aplikace.

To, co činí z tohoto senzoru více než obyčejnou kameru, je především Robotic Vision Core 2 (RVC2). Díky tomuto jádru je OAK schopen zpracovávat výstup z kamer, rozpoznávat objekty, spouštět neuronové sítě a to vše v reálném čase a bez potřeby dalšího externího zařízení. Dále lze využít API DepthAI [26], které dovoluje uživateli konfiguraci a komunikaci se zařízením – podporován je jazyk Python a C++. Celé DepthAI je navíc open-source. Konkrétní využití jsem popsal v kapitole Vytvoření aplikace s podporou hand-trackingu.

### 1.3 Analýza UI

*Do analýzy uživatelského rozhraní jsem zahrnul jak uživatelskou interakci s reálným dronem, tak UI aplikací na bázi hand-trackingu. Pro VR aplikace je UI téměř totožné jako při použití ovladače. Při využití gesture recognition je uživateli často zobrazena zpětná vazba, jaké gesto bylo rozpoznáno.*

Jak jsem již zmínil v předchozí kapitole, uživatel typicky interaguje s dronem pomocí ovladače, který může být dále vybaven obrazovkou. Kromě výstupu z kamery dronu je často tento displej doplněn informacemi o aktuální výšce a rychlosti dronu, stavu baterie, či vzdálenosti od operátora. Existují rovněž speciální FPV brýle, které umožňují uživateli sledovat stream z kamery jako ve virtuální realitě. Pokud dron kameru nemá, musí ho mít operátor neustále na očích.

Jelikož výstupem této práce má být aplikace, obrazovka či brýle pro virtuální realitu jsou v podstatě nutností. V případě virtuálního dronu by bylo možné použít buď pohled z první osoby jako u klasických VR aplikací, pohled z kamery dronu jako v případě FPV brýlí, nebo kameru, která by sledovala pohyb dronu z pohledu třetí osoby. Případně by uživatel mohl mezi těmito možnostmi přepínat.

Co se týče komerčních aplikací využívajících hand-tracking, většina z nich je určena pro VR. Uživatel interaguje s okolím, které reaguje na jeho „dotek“ – od mačkání tlačítek, přes uchopení a přenášení předmětů až po virtuální malování. Tyto funkce hezky ilustruje například aplikace Hand Physics Lab [27]. Uživatelské rozhraní má většinou podobu okna s několika tlačítky, či slidery, na které uživatel klikne přímo prstem. Takovéto menu je buď v horizontální poloze

a „visí“ ve vzduchu, či je umístěno na horizontální plochu. Menu lze obvykle vyvolat tlačítkem, nebo pomocí daného gesta.

Rozpoznání gest probíhá buď v celém viditelném rozsahu senzoru, nebo pouze v ohraničené oblasti – tím lze částečně zamezit zpracování nechtěných gest. Vyhrazený prostor pro interakci bývá na obrazovce zvýrazněn. Zároveň bývá uživateli poskytnuta vizuální zpětná vazba, zda bylo gesto rozpoznáno a jaký má význam. Tento přístup je zvolen především u aplikací, kdy je senzor namířen na uživatele (outside-in). Rovněž je časté promítnutí jednoduché kostry na uživatelovu dlaň, který se tak může přesvědčit, že hand-tracking probíhá správně. Všechny tyto zmíněné vlastnosti ilustruje zmíněná práce [15], která využívá gesture recognition pro ovládání reálného drona.

Senzor může být také umístěn na podložce před uživatelem - jak demonstrují např. v pracích [17] a [28]. V takových případech je uživatel většinou pevně usazen a nemůže se ani příliš otáčet, jelikož by se svou rukou dostal mimo rozsah senzoru. Zároveň může mít senzor problémy s rozpoznáním gesta, když se dlaň uživatele začne přibližovat poloze, kdy je kolmá na podložku. Řešením může být propojení více senzorů [6].

Další možností je svázat senzor s pohybem uživatele a umístit ho například na HMD, jako v případě headsetu od HTC rozšířeného o Leap Motion Controller. Toto řešení umožňuje uživateli volnější pohyb a rotaci těla. Jelikož je pohyb senzoru spojen s pohybem hlavy, může nastat problém při velkém záklonu i rotaci hlavy, kdy se mohou ruce dostat mimo záběr.

Závěrem bych rád zmínil, že mnoho prací a aplikací využívajících hand-tracking dává typicky možnost využít na určité akce i ovladač, klávesnici, či jiné pomocné zařízení pro ovládání. Většinou se jedná o první fázi určité inicializace – v našem případě, řekněme, vzlet dronu – a následně o konečnou fázi – například přistání dronu. Existují však i řešení, kde lze téměř volně střídát ovládání rukou a ovladačem (viz. Oculus Quest). Tato možnost tak poskytuje uživateli „manual override“ nad hand-trackingem a může působit jako jakási zadní vrátka.

## 1.4 Vytvoření aplikace s podporou hand-trackingu

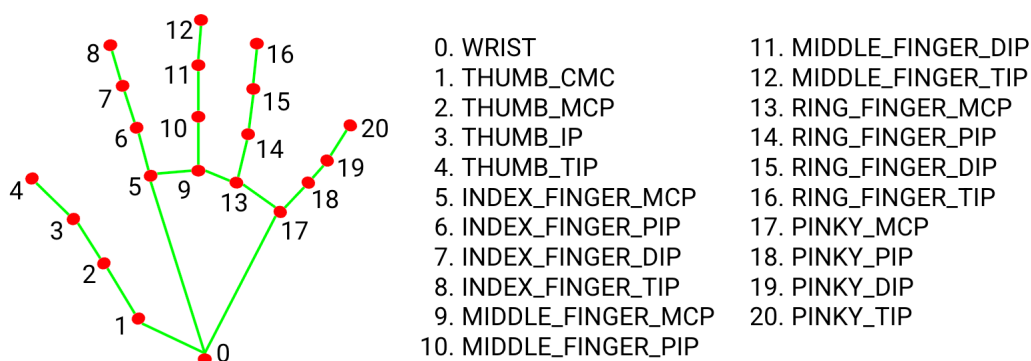
*Cest, jak vytvořit rukou ovládanou aplikaci, je více. V této kapitole zmiňuje nástroje, které vývoj umožňují, či usnadňují. Jedná se především o knihovny, zaměřující se na zpracování obrazu, či rozšíření aplikace Unity. Pro samotný OAK existuje jak API, tak Unity plugin, stejně tak pro Ultraleap.*

Základem pro snímání rukou uživatele je samozřejmě senzor, který je připojen ke stroji, jež zpracovává výstup ze senzoru. Tato výstupní data je typicky potřeba očistit od určitého šumu, který může vzniknout například třesením ruky uživatele. Zároveň nechceme reagovat na každý miniaturní pohyb či rotaci ruky. V pracích, které jsem našel, je tento problém řešen určitým thresholdem, který musí uživatel překonat, aby aplikace na daný pohyb zareagovala – tedy rotace dlaně musí být větší než, dejme tomu, 30 stupňů, aby došlo k pohybu dronu. Jinak aplikace předpokládá, že se jedná o šum a ignoruje jej.

Samotné zpracování obrazu by mělo probíhat v reálném čase. Existují knihovny a frameworky, které to umožňují. Tyto nástroje pomáhají tvůrci aplikace částečně se oprostit od logiky segmentace a dalších potřebných kroků předcházejících samotný hand-tracking a ulehčují tak proces vývoje aplikace. Následuje výčet několika z nich.

### 1.4.1 OpenCV & MediaPipe

Poměrně velkým jménem je v tomto odvětví open source knihovna OpenCV [29]. Nabízí rozhraní pro C++, Python nebo Javu. OpenCV implementuje nespočet funkcí pro počítačové vidění, detekci a trackování objektů, ale i rozpoznání gest a to vše v reálném čase. Její použití je stejné



■ **Obrázek 1.3** MediaPipe model dlaně s prsty mapovaný na dlaň uživatele [32]

přímočaré jako u každé jiné knihovny – stačí ji stáhnout, importovat do projektu a libovolně použít.

Obdobné funkcionality nabízí i knihovna MediaPipe [30], která je rovněž open source. Většina implementací hand-trackingu, které jsem našel, využívá právě kombinaci knihoven OpenCV a MediaPipe. OpenCV se typicky stará o zaznamenání obrazu a jeho zobrazení uživateli, zatímco MediaPipe má na starost trackování. To funguje následovně: na celý snímek se provede Palm detection, jejíž výsledkem je výřez samotné dlaně. Následuje Hand landmarks identification, během které se lokalizují orientační body a mapuje se na ně model s prsty [31]. MediaPipe využívá model s 21 body, typicky se jedná o klouby, koncečky prstů apod. (viz. obrázek 1.3).

Jako ukázka implementace může sloužit práce studentů univerzity v Bandungu, Indonésie [33], kteří s využitím zmíněných knihoven vybudovali gesture recognition system, pomocí kterého lze ovládat jednoduchou aplikaci. Celkem dokázal systém rozpoznat 10 různých gest – díky tomu mohl uživatel listovat v menu bez použití klávesnice či myši. Jako senzor využili Microsoft Kinect.

## 1.4.2 Unity OpenXR & Ultraleap Plugin

Pozadu nezůstal ani systém Unity. Unity je multiplatformní herní engine pro vývoj aplikací pro PC, konzole i mobilní zařízení. Základní verzi softwaru lze rozšířit o množství balíčků, které přináší další funkcionality. Jedním z těchto balíčků je i OpenXR Plugin – nutno dodat, že je zatím ve verzi pre-release.

OpenXR je standard vyvíjený skupinou Khronos Group a má co dočinění především s virtuální a rozšířenou realitou [34]. Jejich balíček pro Unity přináší mimo jiné i podporu hand-trackingu a to nezávisle na platformě – doposud bylo ovládání bez ovladače možné pouze po integraci specifického balíčku (Oculus package, Hololens package, ...). OpenXR slibuje standardizaci a sjednocení. Samotné trackování je však stále úzce svázáno se samotným HMD, jehož součástí je typicky senzor pro snímání pohybu rukou – viz např. Oculus Quest 2.

Pro uživatele, kteří headset nevlastní, je tu pak Ultraleap Plugin pro Unity [35]. Ten umožňuje uživateli využít Leap Motion Controller. Tento balíček přináší funkcionality, na které jsou uživatelé Controlleru zvyklí – vizualizace rukou v reálném čase, kolize s objekty ve virtuální scéně, interakce s objekty apod.

Plugin obsahuje tzv. Unity prefabs, díky kterým lze hand-tracking přímočaré využít pro VR i pro desktop verzi. Při spuštění aplikace se provede namapování a přeškálování modelu na dlaň uživatele (rigging). Lze využít přiložené, optimalizované modely, nebo si vytvořit vlastní. Balíček dále implementuje API pro podrobnou práci s modelem ruky – pomocí něj lze získat informace o jednotlivých prstech, kloubech, pozici ruky, či celých gestech [35].

### 1.4.3 DepthAI & OAK for Unity

Rovněž tvůrci senzoru OAK-D-Lite vytvořili vlastní knihovnu, čímž zpřístupnili potenciál této chytré kamery. Díky tomuto API mají vývojáři přístup k celé škále funkcionalit, které OAK nabízí – od detekce objektů, přes identifikaci obličejů, až po námi kýžený hand-tracking. Na hand-trackingu se podílí již zmíněná knihovna MediaPipe. Proces detekce a mapování tak probíhá obdobně, jako je popsáno výše a model ruky odpovídá obrázku 1.3. Knihovna má distribuce pro operačními systémy Windows, Linux, i macOS.

DepthAI implementuje dva mody, v závislosti na tom, kde probíhají výpočty a zpracování obrazu: Host mode a Edge mode. V případě Host mode probíhá valná většina výpočtů na straně hostitele, typicky PC. Výjimku tvoří neuronová síť běžící přímo na samotném senzoru. Výrobce doporučuje používat Host mode především pro případy, kdy obraz pochází z externího zdroje, např. video soubor, či stream z externí kamery. Zároveň uvádí, že při výpočtech na straně hosta dochází k poklesu FPS. V případě Edge mode jsou prováděny výpočty na samotném zařízení OAK, které je tak logicky nutné mít připojené k host zařízení. Sám výrobce uvádí, že tato varianta je rychlejší a při použití OAK-D-Lite lepší [36].

Dále je možné volit mezi Solo mode a Duo mode – ten určuje počet rukou, které budou senzorem snímány. Solo mode umožňuje hand-tracking pouze jedné ruky. V případě, že jich kamera vidí více, ohodnotí každou ruku interním skóre a rozhodne se pro tu perspektivnější. Nezáleží při tom, zda se jedná o levou, či pravou ruku. Duo mode je pak určen pro trackování dvou rukou, respektive levé a pravé ruky. Pokud vidí pouze jednu ruku, je nucen po určitou dobu provádět detekci druhé ruky, což senzor výrazně zatěžuje. Tuto dobu hledání je možné nastavit pomocí parametru v závislosti na účelu výsledné aplikace. Obecně však platí, že v Duo mode pracuje senzor nejrychleji, pokud neustále vidí obě ruce uživatele – obdobně pro Single mode, kde by měl vidět pouze jednu [37].

Kromě samostatné knihovny existuje i plugin OAK for Unity. Jak název napovídá, jedná se o rozšíření, které umožní vývojářům používat zařízení OAK spolu se softwarem Unity. Zároveň jsou k dispozici předpřipravené projekty, které mohou sloužit jako odrazový můstek pro vývoj aplikace s podporou face recognition, object detection a podobně. Zatím je však tento projekt pouze v alpha verzi a spustitelná demo jsou tak jedním z mála veřejně dostupných výstupů. Hand-tracking navíc v současné době není podporován, avšak v budoucnu by měl být plugin o tuto funkcionalitu rozšířen [38].

## Kapitola 2

# Návrh

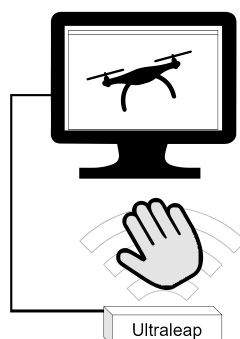
Na základě analýzy jsem navrhl aplikaci pro účely ovládní virtuálního dronu za pomoci hand-trackingu. Návrh počítá s využitím Ultraleap Controlleru pro samotný hand-tracking, pro vývoj se pak hodí nástroj Unity a rozšíření Ultraleap Plugin for Unity. Dále jsem se v návrhu zabýval ovládním pohybem a rotací ruky, následně scénou, ve které se bude dron pohybovat a konečně uživatelským rozhraním aplikace.

### 2.1 Technologie

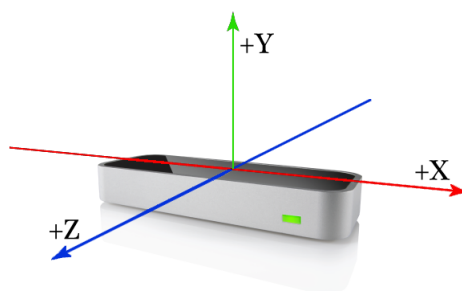
Pro další vývoj desktopové aplikace jsem zvolil engine Unity. Analýza ukázala, že senzor OAK-D-Lite spolu s Unity pro hand-tracking nelze využít. Návrh tedy počítá s využitím Ultraleap Controlleru, který tuto možnost nabízí.

Původním záměrem bylo použít pro účely této diplomové práce senzor OAK-D-Lite. Během analýzy se však ukázalo, že v současné době to není možné. Plugin Oak for Unity sice přináší do prostředí Unity aplikace řadu funkcionalit, které jsou pro senzor zásadní, jako je detekce objektů, detekce obličeje a nálady apod, hand-tracking však zatím podporován není. Na fóru [39] vývojáři zmiňují, že na podpoře hand-trackingu aktivně pracují, ale datum vydání zatím známo není. Rovněž oficiální datum vydání pluginu jako takového zatím nebylo stanoveno.

Dále jsem prozkoumal možnost využití Python wrapperu. Myšlenka byla využít OAK API DepthAI, které běží v Pythonu a plugin Python for Unity [40]. Aplikace by pak posílala data



■ Obrázek 2.1 Zapojení senzoru Ultraleap



■ **Obrázek 2.2** Souřadnicový systém senzoru Ultraleap [41]

ze senzoru do Unity, které by data zpracovávalo a dále s nimi pracovalo. Ani to se však neukázalo jako cesta vpřed. Plugin totiž pracuje pouze s verzí Pythonu 2.7, zatímco DepthAI vyžaduje minimálně verzi 3.0. Zaměnit Unity za Unreal Engine rovněž není možné, jelikož OAK podporu pro tento engine nenabízí.

Padlo tedy rozhodnutí změnit senzor. Na základě analýzy jsem se rozhodl pro senzor Ultraleap Stereo IR 170, který se na hand-tracking specializuje a nabízí podporu jak pro Unity, tak pro Unreal Engine. Jelikož v Unity jsem již v minulosti pracoval, rozhodl jsem využít právě tento software pro vývoj finální aplikace. O komunikaci mezi Unity a senzorem Ultraleap se stará Ultraleap Unity Plugin [35]. Návrh dále počítá s využitím těchto zmíněných technologií.

## 2.2 Ovládání

*Návrh ovládání aplikace je založen na optickém snímání uživatelské ruky. Senzor sleduje pohyb a rotace uživatelské dlaně a reaguje pohybem virtuálního dronu. Zásadní bude odfiltrovat nechtěné pohyby ruky a správně nastavit citlivost, s jakou bude aplikace na pohyby reagovat. Dále jsem navrhl speciální gesta, kterými může uživatel let zahájit, pozastavit a ukončit.*

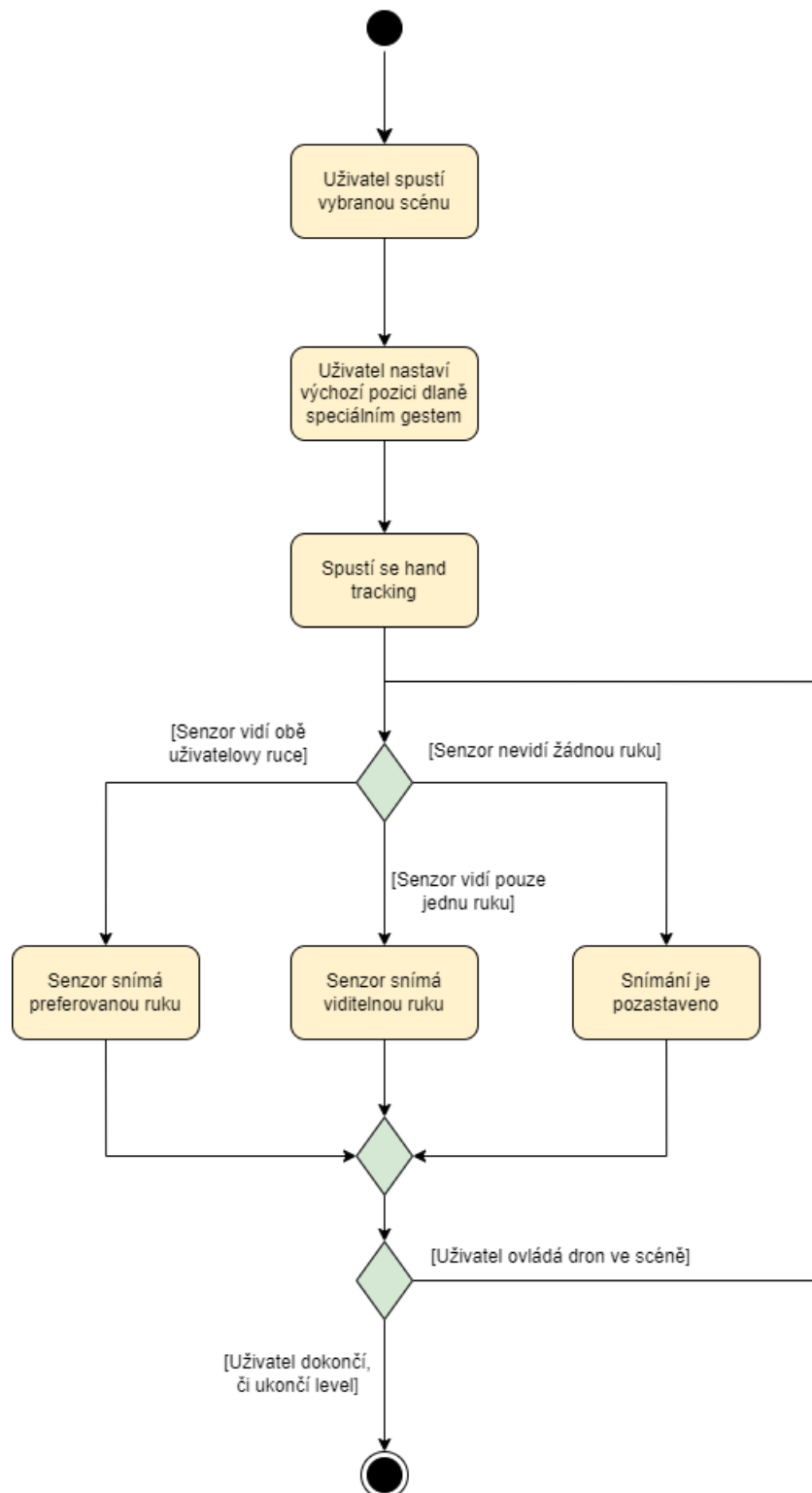
Jelikož se tato diplomová práce zaměřuje na ovládání pomocí hand-trackingu, bude právě tento způsob ovládání pro výslednou aplikaci zásadní. Kromě jednoduchého úvodního menu, kde bude k ovládání sloužit klávesnice a myš, bude ovládání založené na snímání uživatelské dlaně pomocí senzoru Ultraleap.

Senzor by měl být umístěn na stole či jiné vodorovné ploše a mířit směrem vzhůru – jako na obrázku 2.2. Nad senzor umístí uživatel svou dlaň – pouze jednu. I v případě, že senzor zaznamená více dlaní, bude reagovat pouze na jednu. Uživatel si bude moci zvolit preferovanou ruku (levou/pravou), která bude mít v takovém případě „přednost“. Pokud senzor nerozpozná žádnou dlaň, upozorní na tuto skutečnost uživatele, případně pozastaví celou aplikaci. Dlaň nad senzorem bude směřovat směrem dolů, tedy, řečí souřadnicového systému Ultraleap, ve směru  $-Y$ .

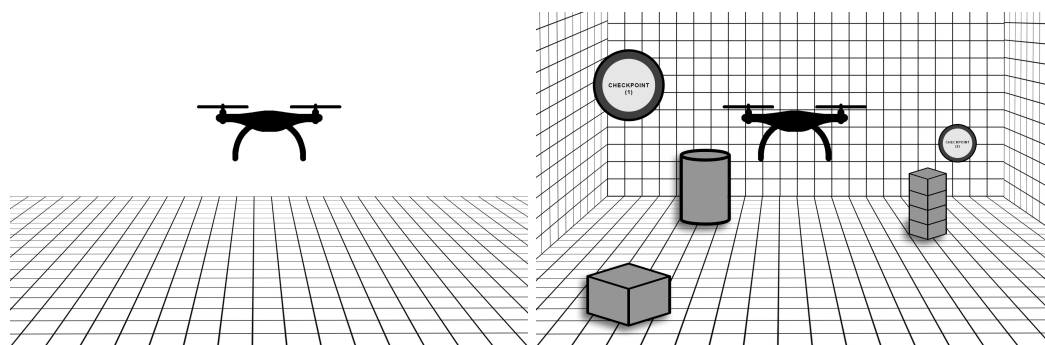
Před zahájením ovládání dronu pomocí hand-trackingu je potřeba zafixovat výchozí pozici dlaně. Je to především kvůli ovládání výšky letu – vychýlení dlaně nad výchozí pozici způsobí stoupaní dronu, naopak vychýlení dlaně pod výchozí pozici způsobí klesání. Pro toto zafixování je potřeba konkrétní, zřetelné gesto – například sevření dlaně v pěst. Minimální vzdálenost, na kterou je zvolený Ultraleap Stereo IR 170 schopen dlaň rozpoznat, je 10 cm, maximální pak 75 cm. Pokud by uživatel zvolil výchozí pozici mimo tuto rozteč, nebo příliš blízko jejím hranicím, mohlo by být ovládání výšky ztíženo, až nemožné. Aplikace proto uživatele na tyto omezení upozorní a vybídne ho k opravě.

Pohyb dlaně ve směru  $Y$ -ové osy tedy bude ovlivňovat výšku. Pohyb dlaně v rovině os  $X$  a  $Z$  nebude mít na pohyb dronu žádný vliv. Pro pohyb dronu dopředu, dozadu, vlevo a vpravo bude sloužit náklon ruky – dron bude vždy následovat směr, ve kterém je dlaň nakloněna. Dron je přitom natočen stále stejným směrem, tedy nerotuje okolo  $Y$ -ové osy.





■ **Obrázek 2.3** Proces snímání rukou uživatele za běhu aplikace



■ **Obrázek 2.4** Návrh prázdné tréninkové scény (vlevo) a jednoduché scény s několika překážkami a checkpointy (vpravo)

Změnu směru letu, a tedy rotaci dronu kolem své osy, způsobí rotace ruky kolem Y-ové osy. Dlaně je přitom stále v rovině a nerotuje kolem X-ové, ani Z-ové osy. Tento pohyb lze vykonat pouze zápěstím, nebo rotovat celou rukou.

Dále jsem v rámci ovládání navrhl speciální gesto, kdy uživatel otočí ruku dlaní vzhůru – dron reaguje tak, že provede otočku o  $360^\circ$  v Z-ové ose. Směr otočení přitom bude záležet na tom, kterou rukou uživatel gesto provedl. V případě pravé ruky provede dron otočku doprava a naopak. Všechny výše zmíněné ovládací prvky shrnuje tabulka.

Ovládání je navrženo tak, že míra vychýlení z výchozí pozice odpovídá rychlosti pohybu dronu. Pokud tedy uživatel zvedne ruku, dejme tomu, o 30 centimetrů, bude dron stoupat rychleji, než když ji zvedne pouze o 10 centimetrů. To samé platí pro náklon a rotaci dlaně – větší náklon znamená rychlejší pohyb dronu, respektive rychlejší rotaci.

Tím, že je rozsah pohybu dlaně omezený, bude potřeba jednotlivé pohyby a náklony škálovat a experimentálně najít takové meze, při kterých bude ovládání pro uživatele co možná nejpřirozenější. Toto nastavení citlivosti by mohl uživatel změnit v menu. Dále bude během implementace důležité zaměřit se na čištění dat od šumu, který může vzniknout například roztřesením ruky. Šum lze odstranit kupříkladu určitou hranicí (thresholdem), kterou musí pohyb překročit, aby byl zaznamenán a dron na něj reagoval.










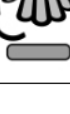
## 2.3 Scéna

*Dron se nebude pohybovat pouze v prázdném prostoru, nýbrž ve vytvořených scénách. V této kapitole jsem se zabýval jejich strukturou, vytvářením a případným generováním. Zároveň zde rozebírám myšlenku gamifikace celé aplikace, kde uživatelským cílem může být co nejrychlejší, či co nejpřesnější průlet scénou.*

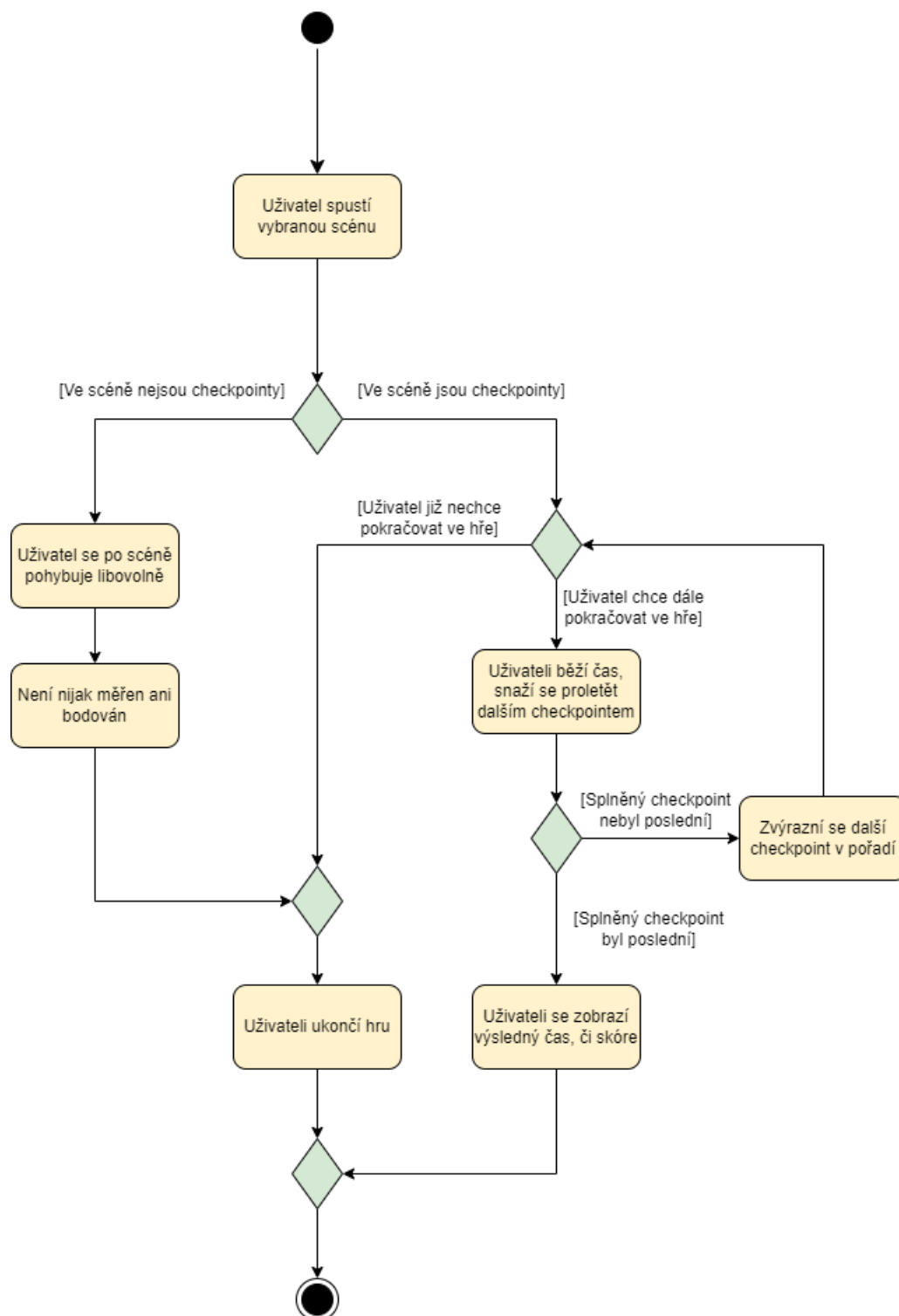
V předchozí kapitole jsem objasnil, jak se bude dron pohybovat. V této kapitole rozeberu, kde se bude pohybovat. Jak již bylo řečeno, vývoj aplikace bude probíhat v prostředí softwaru Unity. Nabízí se tak využít potenciál pro tvorbu 3D scén, kterým tento herní engine disponuje.

Let dronem po prázdné, trojrozměrné scéně uživatele pravděpodobně nijak neohromí, ani příliš nezabaví – přesto, že bude dron ovládat vlastním tělem. Může to však sloužit jako vhodný výchozí bod právě pro seznámení se specifickým ovládáním. Tato výuková scéna může být buď „nekonečným“ prostorem, kde jediným záchytným bodem bude podlaha, nebo uzavřenou, dostatečně velkou krychlí, kterou nebude možné opustit.

Poté, co si uživatel osahá ovládání pomocí hand-trackingu v úvodní scéně, může přejít k zajímavějším scénám. I díky vývoji v Unity se nabízí gamifikace celé aplikace. Do scény by mohly být umístěny průletové body, kterými by musel uživatel v daném pořadí proletět, aby úroveň splnil. Tyto checkpointy mohou mít tvar koule, či kruhu. Průlet by byl zaznamenán, po-

GESTO	obrázek	AKCE
Sevření dlaně v pěst		Zafixování výchozí pozice
Plochá dlaň ve výchozí pozici		Dron se nehýbe (hovering)
Vychýlení dlaně z výchozí pozice směrem vzhůru		Dron stoupá
Vychýlení dlaně z výchozí pozice směrem dolů		Dron klesá
Náklon dlaně doleva (v Z-ové ose)		Dron letí doleva
Náklon dlaně dopředu (v X-ové ose)		Dron letí vpřed
Náklon dlaně dozadu (v X-ové ose)		Dron letí dozadu (couvá)
Rotace dlaně v Y-ové ose doleva		Dron se otáčí doleva
Rotace dlaně v Y-ové ose doprava		Dron se otáčí doprava
Otočení dlaně směrem vzhůru		Dron udělá otočku

■ **Tabulka 2.1** Návrh ovládání pomocí hand-trackingu



■ Obrázek 2.5 Návrh gamifikace aplikace

kud by se dron alespoň dotkl vymezeného tělesa. Případně lze laťku posunout výše a požadovat, aby tělesem proletěl celý dron. Takto splněný checkpoint ze scény zmizí a uživateli se zobrazí text o úspěšném průletu – tím dostane zpětnou vazbu, že může se soustředit na další průletový bod.

Do scény lze dále doplnit překážky, aby byl průlet obtížnější. Uživatele lze hodnotit například celkovým časem, který pro splnění úrovně potřeboval. Čas se spustí při spuštění úrovně. Tím, že pořadí checkpointů je dané, bude vždy jeden z nich tím posledním. V moment, kdy jím uživatel proletí, se čas zastaví. Uživatel se logicky snaží proletět úrovní co nejrychleji, tedy v co nejkratším čase.

Dalším možným kritériem hodnocení je spojení času a přesnosti. Uživatel by sbíral body při každém průletu checkpointy. Checkpoint si pak můžeme představit jako terč, kde nejvyšší bodové ohodnocení je v jeho středu a směrem ke krajům klesá. Body za průlet by pak uživatel obdržel v závislosti na vzdálenosti od středu a čase mezi průlety. Opět by platilo čím kratší čas, tím vyšší bodový zisk. Dále lze přidat bonusové body za okamžitou rychlost dronu v moment průletu – tady by platila přímá úměra.

Výsledný bodový zisk z celé úrovně je pak součtem bodů za jednotlivé průlety, případně celkový čas. Nabízí se toto skóre ukládat a udržovat lokální tabulku s nejvyšším skóre, respektive s nejrychlejšími časy. To rovněž napomůže znovuhratelnosti.

Pro lepší orientaci ve scéně bude potřeba nějaký ukazatel, který jasně určí, kterým checkpointem má uživatelem proletět jako dalším. Checkpoint lze například barevně odlišit, přidat na obrazovku šipku, která k němu bude směřovat, nebo přidat vizualizaci nejkratší dráhy letu k danému checkpointu. Dále lze přidat na obrazovku minimapu, která uživateli napoví, kam letět. Těmto aspektům se podrobněji věnuji v následující kapitole 2.4.

3D scény, či úrovně, které bude uživatel zdolávat, lze vytvářet v prostředí Unity ručně. Bylo by však možné je i procedurálně generovat. To však není cílem této diplomové práce a může to být předmětem dalších prací, které tuto problematiku dále rozvedou.

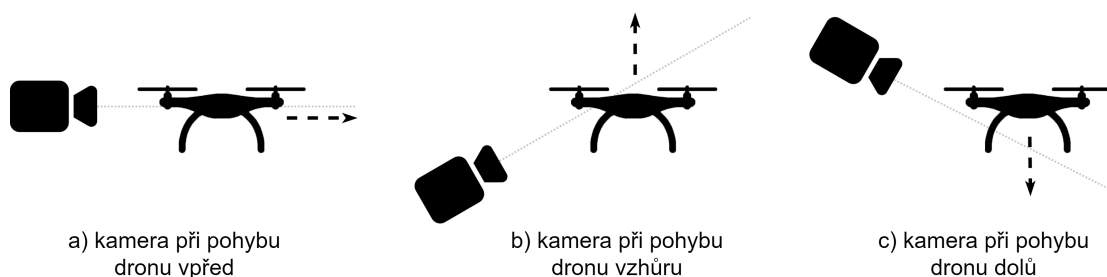
## 2.4 Návrh UI

*V této části jsem se zabýval návrhem uživatelského rozhraní aplikace. Uživatel bude dron sledovat z pohledu třetí osoby, kamera pak bude logicky kopírovat pohyb a natočení dronu. Dále budou uživateli zobrazeny informace o letu a také pohled z pozice senzoru, aby se mohl ujistit, že hand-tracking probíhá správně. Nakonec jsem navrhl jednoduché menu aplikace.*

Tato práce se soustředí především na implementaci ovládání virtuálního dronu za pomoci technik hand-trackingu. Dron je proto v návrhu logicky středobodem obrazovky. Na dron nahlíží kamera z pohledu třetí osoby. Kamera je přitom zafixovaná na pohyb a rotaci dronu. Uživatel tak vidí jak dron, který ovládá, tak scénu před ním.

Jak již zaznělo, kamera kopíruje pohyb dronu ve všech směrech. Rotaci však kopíruje pouze ve směru osy Y, která směřuje směrem nahoru. Pokud se tedy dron otáčí kolem své osy, otáčí se i kamera. Pokud se však dron nakloní při pohybu do stran, kamera zůstane stále vodorovně a pouze kopíruje pohyb vlevo/vpravo. Obdobná situace nastává při pohybu dronu vpřed a vzad – kamera následuje dron, avšak nerotuje kolem osy X a opět zůstává vodorovně. Jinými slovy, kamera se neustále dívá přímo vpřed a zůstává rovnoběžná na rovinu XZ.

Výjimku by mohl tvořit pohyb dronu směrem vzhůru a dolů. V takovém případě sice dron nijak nerotuje, ale mohla by rotovat samotná kamera. Pokud dron klesá, mohlo by být pro uživatele příjemnější, aby viděl i co je pod ním, namísto pohledu jen přímo před sebe. Kamera by se tak mohla částečně natočit směrem dolů. Obdobně při stoupání dronu, kdy by kamera byla natočena nahoru a pohled by zabíral jako prostor před, tak i nad dronem. Obě situace zachycuje obrázek 2.6. Tento dynamický pohyb kamery by opět bylo možné v menu aktivovat a deaktivovat. Případně by podobný princip mohl být aplikován i při pohybu dronu do stran, či pokud by dron couval.



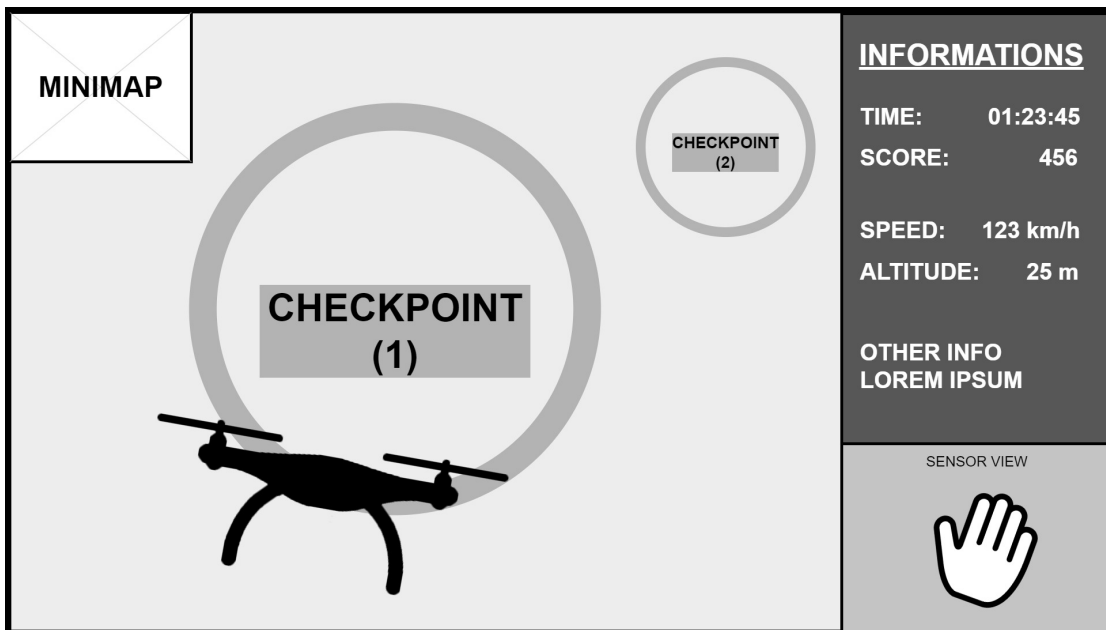
■ **Obrázek 2.6** Návrh dynamického pohybu kamery

V případě gamifikace by byly ve scéně rozmístěny checkpointy, kterými by se snažil uživatel proletět v co nejkratším čase. Pro uživatele tak bude zásadní uplynulý čas, či skóre, dále pak výška letu, rychlost letu, případně pak další informace o průletech checkpointy apod. Tyto informace jsou v návrhu umístěny v pravé části obrazovky. Pokud by se však uživatel chtěl soustředit pouze na samotný let, mohl by tento sidebar skrýt, čímž by se mu i částečně rozšířilo zorné pole. V tomto „fullscreen modu“ je však stále přítomný ubíhající čas, který je pro uživatele zásadní.

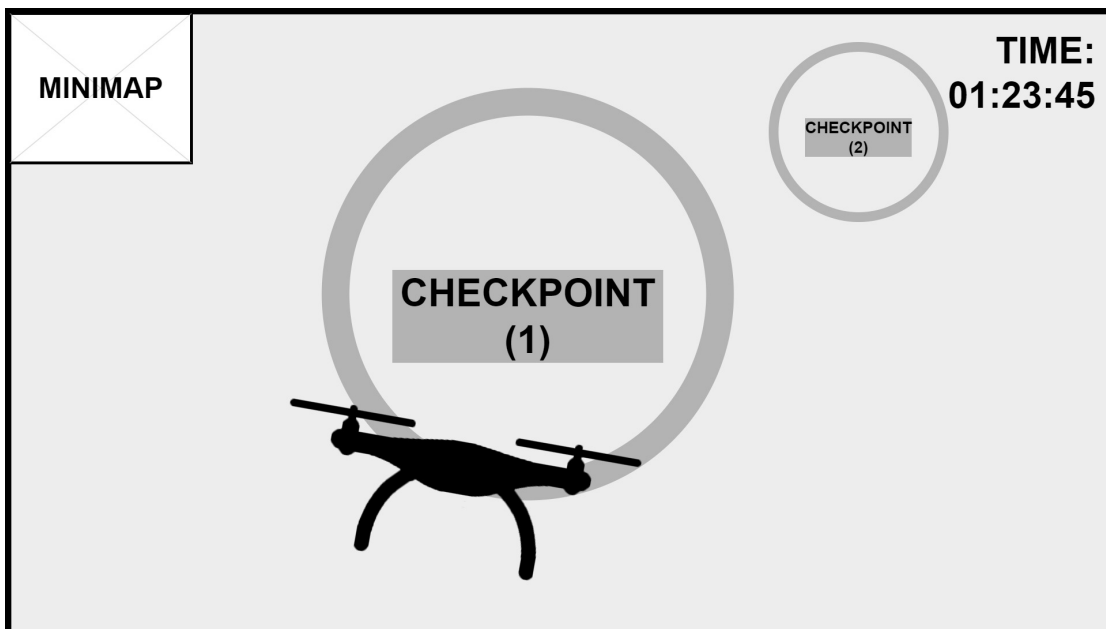
Součástí zobrazitelného sidebaru s informacemi je rovněž pohled z pozice senzoru směrem vzhůru, tedy na uživatelovy dlaně. Slouží především k tomu, aby se uživatel ujistil, že hand-tracking probíhá správně, a že se jeho dlaně pohybují ve vzdálenosti, kterou je senzor schopen snímat a zpracovat.

Dalším důležitým údajem je pozice dronu ve scéně. Pro lepší orientaci jsem tak v návrhu umístil do levého horního rohu minimapu, která by uživateli měla usnadnit navigaci k dalšímu cíli. Minimapa bude zjednodušeně pohled na scénu shora. Z minimapy uživatel vyčte jak svou pozici, tak pozici jednotlivých checkpointů – včetně toho následujícího. Jelikož se dron bude pohybovat po trojrozměrném prostředí, je potřeba do 2D minimapy umístit indikátor, který uživateli napoví, zda je checkpoint nad ním, či pod ním. K tomu by mohla sloužit jednoduchá šipka vedle ikony checkpointu, která by směřovala vždy buď nahoru, či dolů a to vždy podle relativní pozice vůči dronu.

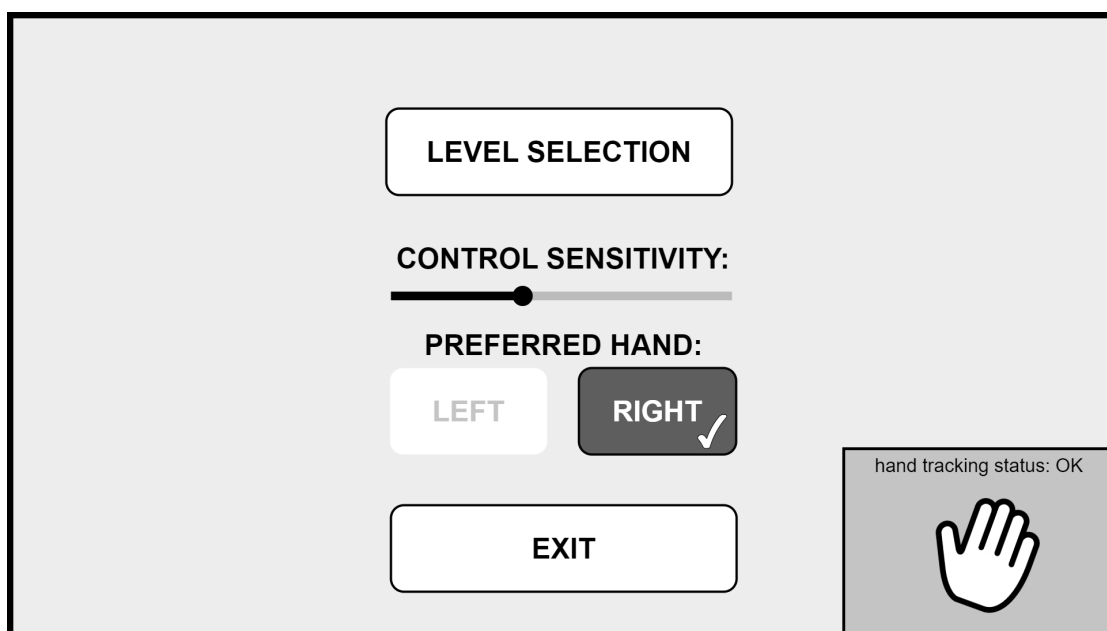
Nakonec jsem navrhl jednoduché menu, které by v případě gamifikace aplikace fungovalo zároveň jako úvodní obrazovka po spuštění. Z menu by si mohl uživatel zvolit scénu (level), kterou by chtěl spustit. Mohl by si zvolit buď již zmíněnou prázdnou, tréninkovou scénu, či jednu ze scén s překážkami a checkpointy. Dále se v menu nachází nastavení citlivosti ovládání a volba preferované ruky (levá/pravá). V pravém dolním rohu obrazovky je umístěn pohled z pozice senzoru, který zároveň signalizuje, zda je Ultraleap Controller připojen a hand-tracking probíhá správně. Nakonec je zde tlačítko pro ukončení aplikace.



■ Obrázek 2.7 Návrh UI se sidebarem s informacemi o letu



■ Obrázek 2.8 Návrh UI ve fullscreen modu



■ Obrázek 2.9 Návrh jednoduchého menu



# Implementace

*V této kapitole popisují samotnou implementaci výsledné aplikace v Unity s využitím Ultraleap Controlleru. O komunikaci mezi senzorem a aplikací se stará Ultraleap Plugin for Unity. Ovládání prošlo od návrhu několika změnami, tou nejzásadnější je navázání rychlosti pohybu na počet zdvižených prstů. Aplikaci jsem proměnil v jednoduchou hru, ve které je uživatelským cílem proletět checkpointy v co nejkratším čase. Dále jsou v této kapitole popsány problémy, na které jsem při implementaci narazil a možnosti budoucího rozšíření.*

## 3.1 Předání a zpracování dat

*Pro přístup k datům ze senzoru a jejich zpracování je zásadní třída `Leap::Frame` [42]. Ta nese informace o detekovaných dlaních v každém snímku a další metadata. Samotná data se předávají pomocí třídy `Leap::Controller` [43], která slouží jako interface mezi senzorem a pluginem. `LeapProvider` [44] pak poskytuje potřebná data k dalšímu zpracování, jakmile jsou k dispozici. O samotné zpracování dat se stará třída `MyGameManager`.*

Jak jsem již zmínil v návrhu, o předání dat mezi Unity a senzorem Ultraleap Stereo IR 170 se stará Ultraleap Plugin for Unity, konkrétně verze 6.12.1. Vývoj pak probíhal v Unity verze 2021.3.11f1 Education. Aby do Unity proudily data ze senzoru, stačí ho připojit k danému zařízení a do scény umístit takzvané prefab – naskriptované objekty připravené k použití v Unity. Konkrétně prefab Service Provider, který se stará právě o předání dat a jeden z připravených modelů rukou, které stínují reálný pohyb rukou uživatele.

Tím je v podstatě vyřešen samotný proces detekce rukou. Zbývá zpracovat data, které senzor nasbírá v každém snímku, přičemž maximální snímkovací frekvence námi zvoleného senzoru je 90 FPS [45]. Data, s kterými pak program dále pracuje, nese třída `Frame`. Každý `Frame` je obtiskem toho, co zrovna v daný snímek senzor zaznamenal a obsahuje:

- `id` – jedinečný identifikátor pro daný `Frame`
- `timestamp` – čas, kdy byl snímek pořízen
- `fps` – aktuální snímkovací frekvence
- `hands` – seznam všech detekovaných rukou

Právě poslední položka `hands` je pro nás zásadní. Každá detekovaná ruka je instancí třídy `Hand` [46], která popisuje fyzické charakteristiky ruky, stejně jako její umístění v prostoru. Lze tak zjistit relativní pozici dlaně vůči senzoru, její směr, rychlost, úhly jednotlivých prstů vůči

dlaní, či další podrobnosti potřebné k rozpoznání tvaru dlaně a pozice prstů. Samozřejmostí pak je, že senzor pozná, zda se jedná o levou, či pravou ruku.

Další důležitou třídou je LeapProvider, který funguje jako distributor těchto dat. Ostatní třídy se přihlásí k odběru informací a Provider jim je každý snímek poskytne. Jedním z těchto odběratelů je i třída MyGameManager, která mimo jiné převezme informaci o ruce, které senzor právě vidí a připraví je k dalšímu zpracování. Tato data musí být přístupná a jednotná, tudíž musí být v celé aplikaci pouze jedna instance této třídy – je zde proto použit návrhový vzor singleton. Pokud senzor vidí více rukou, zpracuje pouze tu preferovanou (levou/pravou), spočítá počet zdvižených prstů apod. – více v kapitole Ovládání. Takto předzpracované informace se předají dále třídám, které se starají o pohyb dronu v závislosti na pozici a rotaci ruky uživatele.

Třída MyGameManager dále udržuje důležité informace o stavu a nastavení hry – zda je uživatel v menu, zda má správně nastavenou výchozí pozici dlaně, zda již level dokončil a tak dále. Rovněž si drží reference na další objekty (menu, checkpointy apod.). Částečně se také stará o prvotní nastavení scény, kdy aktivuje právě první checkpoint a nastaví na něj ukazatel směru.

## 3.2 Ovládání

*Ovládání dronu pomocí hand-trackingu je navázáno na pozici, rotaci a náklon uživateli dlaně. Rychlost pohybu vpřed je přímo úměrná počtu zdvižených prstů, zpátečka má jen jednu rychlost. Pro ovládání výšky letu je potřeba gestem pěsti zafixovat výchozí pozici dlaně. Pokud je dlaň nad touto rovinou, dron stoupá – obdobně pro klesání. Kromě ovládání hand-trackingem jsem naimplementoval i jednoduché ovládání pomocí klávesnice.*

Implementace ovládání letu dronu pomocí hand-trackingu byla jedním z hlavních cílů této práce. Specifika jednotlivých pohybů jsou popsány v podkapitolách, nejprve však několik obecných slov úvodem.

Každý objekt v softwaru Unity, který je umístěn do scény, má parametr třídy Transform [47], který ovládá jeho pozici, rotaci a měřítko. Právě manipulací atributů tohoto parametru je docíleno pohybu a rotace dronu. Atributy jsou změněny každý snímek (frame), vždy v závislosti na pohybu uživatele dlaně. Aby nebyl pohyb závislý na snímkovací frekvenci (FPS), je míra změny přenásobena atributem Time.deltaTime [48], který udržuje počet sekund mezi posledním a současným snímkem.

Aby dron skutečně reagoval na pohyb dlaně, musí být pohyb dostatečně znatelný. Pro každý pohyb, náklon a rotaci jsem tak nastavil určitou hranici, kterou je třeba překonat, aby byl pohyb zaregistrován. Hodnoty těchto thresholdů byly předmětem dílčího testování během vývoje. Zároveň tímto opatřením vzrostla robustnost ovládání.

Při implementaci jsem myslel na levoruké i pravoruké uživatele. Pro ovládání si tak lze zvolit preferovanou ruku, která bude mít přednost v případě, že senzor uvidí levou i pravou ruku uživatele. Toto nastavení lze kdykoliv změnit v menu. Pokud senzor vidí pouze „nepreferovanou“ ruku, snímá právě tu.

### 3.2.1 Pohyb dopředu a dozadu

Od návrhu prošlo ovládání několika změnami. Původně měl být pohyb vpřed navázán na náklon dlaně dopředu, přičemž míra náklonu by určovala rychlost letu. Dlaň ve vodorovné poloze by tak znamenala, že dron „stojí“ na místě (hovering), náklon dlaně dozadu by pak znamenal zpátečku. Zdálo se logické, že pohyb dronu vpřed spolu s rotací bude nejčastějším scénářem a bylo tedy potřeba, aby byl pro uživatele co možná nejpřirozenější.

Při průběžném testování jsem si uvědomil, že pokud uživatel nakloní dlaň dopředu tak, že je téměř kolmá na rovinu, na které je umístěn senzor, znemožňuje to částečně možnost rotovat zápěstím. Otočení dronu by tak bylo potřeba vykonat pohybem celé ruky. Navíc pokud je dlaň

v této poloze, má senzor problémy ji rozpoznat a docházelo tak k občasným výpadkům – a to jak při pohybu vpřed, tak i vzad. Naopak pokud je dlaň ve vodorovné poloze, senzor pracuje spolehlivěji a dlaní lze snadněji rotovat.

Padlo tedy rozhodnutí navázat rychlost pohybu na počet zdvižených prstů. Otevřená dlaň ve vodorovné poloze tak znamená pohyb vpřed maximální rychlostí, namísto původního záměru, aby se při tomto gestu dron nehýbal. Náklon dlaně dopředu nyní nemá na pohyb dronu vliv. Náklon dlaně dozadu způsobí couvání dronu, tedy stejně jako v návrhu. Je zde jediný rozdíl v tom, že pro couvání je jednotná rychlost a nezáleží tak na konkrétním náklonu, ani počtu zdvižených prstů – senzor má totiž, jak jsem již naznačil, potíže detekovat jednotlivé prsty, pokud se náklon dlaně přiblíží devadesáti stupňům.

Pro pohyb dopředu tak existuje pět různých rychlostí, pro pohyb dozadu jen jedna – obdobně, jako například v autě. Nezáleží při tom, které prsty jsou natažené, ale pouze na jejich celkovém počtu. Nezávisí tedy, zda má uživatel zdvižený ukazováček a prostředníček, nebo palec a malíček, v obou případech letí dron rychlostí dva.

Pokud uživatel sevře dlaň v pěst, dron se nehýbe, respektive neletí vpřed. Stále může vykonávat pohyb nahoru/dolů, do stran a rotovat. Gesto pěsti je použito i pro zafixování výchozí pozice pro stoupaní a klesání, tu však lze nastavit pouze v menu a při samotném letu tak slouží gesto k zastavení dronu.

### 3.2.2 Pohyb do stran a rotace

Co se týče pohybu do stran, ten zůstal oproti návrhu stejný. Realizuje se náklonem dlaně doprava, respektive doleva. Platí, že rychlost pohybu do strany je přímo úměrná náklonu dlaně. Maximální rychlost má tak dron při náklonu dlaně o devadesát stupňů. Nezáleží přitom na počtu zdvižených prstů, jak tomu je u pohybu vpřed. Pohyb do stran lze vykonat ať už dron stojí, nebo je v pohybu.

I zde je potřeba náklonem dlaně překonat určitou hranici, aby byl pohyb zaregistrován. Po dílčím testování jsem tento minimální úhel nastavil na 45°, maximální úhel je pak zmíněných 90°. Zkoumá se přitom normála dlaně, tedy kolmice na rovinu dlaně. Může se to zdát jako poměrně malý rozsah, ale odfiltruje se tím naprostá většina nechtěných náklonů dlaně. Navíc není přirozené mít dlaň přesně vodorovně, ale mírně nakloněnou – i proto byl rozsah zvolen takto.

Rovněž rotaci dronu jsem naimplementoval tak, jak zmiňuji v návrhu. Vykoná se rotací dlaně v Y-ové ose, tedy v rovině, na které je umístěn senzor. I zde platí, že čím více vychýlí uživatel dlaň, tedy čím více jí natočí doleva, či doprava, tím rychleji dron rotuje. Rotaci lze provést zápěstím, či celou rukou – výsledek je stejný, jelikož senzor sleduje směr natažených prstů (v případě sevřené pěsti se rotace snímá tak, jako by dlaň byla otevřená).

Rotace ruky je zaregistrována, pokud přesáhne zhruba 15° doleva, či doprava. Pokud je úhel menší, dron letí rovně (v případě zdvižených prstů), či stojí na místě (v případě pěsti). Maximální rychlosti rotace pak uživatel docílí rotací dlaně o 90° na danou stranu. I pokud úhel překoná hranici 90°, rychlost rotace se již nezvýší.

Jak pohyb do stran, tak rotaci, lze samozřejmě kombinovat s ostatními pohyby dronu. Typicky uživatel většinu času poletí vpřed a směr letu bude korigovat právě rotací dlaně, případně náklonem. Přitom bude ještě potřeba měnit výšku letu – o tom pojednává následující podkapitola 3.2.3.

### 3.2.3 Pohyb nahoru a dolů

Pro ovládání výšky letu je zásadní zafixování výchozí pozice dlaně. Bez výchozí pozice nelze s dronem stoupat, ani klesat a volba této pozice je tak nutnou prerekvizitou pro využití samotného hand-trackingu. Nastavení této pozice probíhá v úvodním menu, její změna pak rovněž v menu, které lze kdykoliv vyvolat klávesou Escape.

Uživatel provede fixaci tak, že umístí svou dlaň nad senzor a sevře ji v pěst. Aplikace si uloží výšku, ve které uživatel gesto provedl a ta od té doby slouží jako neutrální hladina pro ovládání výšky letu dronu. Pokud je uživatelská dlaň níže, než tato hladina, dron klesá. Obdobně to platí pro stoupání dronu.

Jelikož senzor Ultraleap Stereo IR 170 zvládá rozpoznat ruce v rozsahu 10–75 centimetrů, je nutné zajistit, aby výchozí pozice nebyla příliš blízko těmto hranicím, nebo dokonce mimo ně. Při nastavování tak aplikace uživatele upozorní, že pozici zvolil příliš nízko (méně než 20 cm nad senzorem), nebo příliš vysoko (více než 60 cm nad senzorem) a vyzve ho k opravě. Bez validní výchozí pozice je možné dron ovládat pouze klávesnicí – viz další podkapitola 3.2.4.

I tady platí, že míra vychýlení odpovídá rychlosti stoupání, či klesání. Aplikace tak spočítá rozdíl mezi aktuální a výchozí výškou dlaně a podle něj přeškaluje rychlost. Maximální rychlost je však shora omezena a uživatel jí dosáhne pohybem o zhruba 15 centimetrů nahoru, respektive dolů. Minimální rychlosti, a tedy minimálním zaregistrovatelným pohybem, dosáhne pohybem o 5 centimetrů – okolo výchozí výšky je tak deseticentimetrové pásmo, které odfiltruje nechtěné vertikální pohyby ruky.

### 3.2.4 Ovládání klávesnicí

Vedle implementace ovládání dronu pomocí hand-trackingu, které bylo hlavním cílem této práce, jsem naimplementoval i jednoduché ovládání pomocí klávesnice. Oproti hand-trackingu nenabízí takovou možnost pohybu, konkrétně zde chybí pohyb do stran. Logicky zde odpadává potřeba zafixovat výchozí pozici dlaně pro pohyb nahoru a dolů.

Pohyb dopředu a dozadu je navázán na příslušné šipky na klávesnici, šipky do stran pak zajišťují rotaci dronu. Pro stoupání dronu slouží klávesa mezerník a pro klesání klávesa C. Pokud uživatel nedrží žádnou klávesu, dron se nehýbe. Ovládání klávesnicí nenabízí možnost pohybu různou rychlostí, je zde tedy jednotná rychlost pro pohyb vpřed i vzad. Rovněž rychlost rotace je neměnná. Kvůli tomu působí o něco méně plynule, než ovládání hand-trackingem.

Mezi klávesnicí a hand-trackingem lze libovolně přecházet. Ovládání klávesnicí je pro tuto práci druhotné a není tak příliš optimalizované, ale ani příliš podstatné. Může ale nabídnout jednoduchou alternativu pro uživatele, kteří by si výslednou aplikaci chtěli vyzkoušet i bez senzoru Ultraleap. Zároveň to může být zajímavá možnost vzhledem k testování, kde se lze zaměřit na porovnání těchto dvou způsobů ovládání.

## 3.3 Gamifikace

*Výslednou aplikaci jsem proměnil v jednoduchou hru. Uživatelským cílem je proletět všemi checkpointy ve scéně za co nejkratší čas. Vytvořil jsem jednu výukovou scénu, kde se uživatel seznámí se specifickým ovládáním a další scénu, kde nabyté zkušenosti aplikuje v praxi. Vytváření dalších levelů do budoucna usnadní připravené Unity prefabs.*

Již v návrhu jsem zmiňoval myšlenku o gamifikaci celé aplikace, zde jsem myšlenku realizoval. Uživatel ovládá model dronu [49], kterým se pohybuje po vytvořené scéně. Jeho cílem je proletět v co nejkratším čase průletovými body – checkpointy. Každý checkpoint má tvar koule, která má průměr o něco větší, než je velikost modelu dronu. Aby uživatel poznal, že se skutečně jedná o průletový bod a nikoliv pouze o překážku, která je součástí scény, je nad koulí umístěn rotující nápis „CHECKPOINT“ a celý model má světle zelenou barvu, která navíc emituje světlo. Pro uživatele by tak neměl být problém rozpoznat checkpoint zasazený do prostředí, po kterém se pohybuje.

Ve scéně je vždy aktivní a viditelný pouze jeden checkpoint. Až poté, co jím uživatel proletí, daný checkpoint zmizí a objeví se nový – ten následující. Směr k dalšímu průletovému bodu ukazuje šipka v horní části obrazovky (více v kapitole 3.4 UI aplikace). Počet checkpointů ve scéně je



■ **Obrázek 3.1** Výuková scéna s několika překážkami a popsaným ovládáním

vždy konečný, a čas tak běží od spuštění levelu po průlet posledním checkpointem. Uživatel se přitom snaží, aby byl čas co nejkratší. Pokud chce uživatel čas, a tím i celou hru, pozastavit, může zmáčknout tlačítko Escape, čímž vyvolá jednoduché menu. Čas se znovu rozběhne, když se do hry vrátí.

Pro účely této práce jsem vytvořil jednoduchou výukovou scénu, která neobsahuje checkpointy a kde není nijak měřen čas. Jejím účelem je seznámit uživatele s ovládáním aplikace. Ve scéně je pouze několik překážek a velká tabule, která shrnuje specifika ovládání – a to jak hand-trackingem, tak klávesnicí. Zde si může uživatel vyzkoušet, jak pohyby ruky ovlivňují dráhu letu dronu a poté se přesunout k ukázkovému levelu s checkpointy, kde je mu již měřen celkový čas průletu.

Scénu tohoto levelu jsem nevytvořil sám, ale použil jsem demo scénu z Unity Assetu Simple Town [50]. Po ní jsem rozmístil celkem 15 checkpointů a její průlet by měl uživateli v ideálním případě zabrat asi minutu a půl. Může přitom využít buď ovládání hand-trackingem nebo klávesnicí. Jak proměnit vytvořenou v scénu v použitelný level pomocí vytvořených prefabs pak blíže popisují v příloze B.

### 3.4 UI aplikace

*V této kapitole popisují uživatelské rozhraní aplikace. Pro uživatele je zásadní uplynulý čas, pro navigaci k dalšímu checkpointu pak slouží šipka ukazující směr. Pohyb po scéně usnadní i minimapa. Dále uživatel vidí počet splněných checkpointů a pohled ze senzoru, kde se můžete ujistit a správném fungování hand-trackingu. Závěrem zde zmiňuji ještě dynamickou kameru a jednoduché menu.*

V návrhu jsem původně zamýšlel, že pro let vytvořím dvě obrazovky, mezi kterými bude možné přepínat. Jednu čistší, kde by kromě pohledu na dron byl pouze uplynulý čas a minimapa, a druhou navíc s postranním panelem, kde by byly zobrazeny informace o letu, kontrola hand-trackingu a další informace. Nakonec jsem zvolil kompromis mezi těmito variantami.



■ **Obrázek 3.2** Uživatelské rozhraní aplikace při letu dronem

Jelikož se uživatel snaží proletět scénou v co nejkratším čase, je pro něj důležité vědět, jak si zatím vede. V levém horním rohu tak vidí, kolik z celkových průletových bodů již absolvoval. V pravém horním rohu pak vidí uplynulý čas.

Pro navigaci po scéně je zásadní šipka ukazující směr k dalšímu checkpointu. Ta je umístěna v horní části obrazovky, přímo nad dronem. Uživatel má tak vždy přehled, kterým směrem by se měl vydat. Stále se však musí vyhýbat překážkám ve scéně – šipka míří „pouze“ vzdušnou čarou na pozici dalšího checkpointu a může tak ukazovat i skrze neprůchozí překážky.

V levém dolním rohu je pak umístěna minimapa, která může rovněž napomoci lepší orientaci ve scéně. Minimapa funguje jako ortogonální pohled na scénu shora, ve jejímž středu je ovládaný dron. Kopíruje přitom pohyb i rotaci dronu.

O tom, zda hand-tracking probíhá správně, se může uživatel přesvědčit v pravém dolním rohu obrazovky. Nalezne zde pohled z pozice senzoru a vizualizaci snímaných rukou. Může se tak přesvědčit, zda senzor dokázal jeho ruku rozpoznat, nebo se již ruka nachází mimo snímatelný rozsah.

Všechny výše zmíněné prvky uživatelského rozhraní mají svou pevně danou pozici na obrazovce a jsou přítomné, ať se dron nachází kdekoliv ve scéně. Nejzásadnější je pak samozřejmě samotný vhled do scény. Ten nabízí kamera, která sleduje dron z pohledu třetí osoby. Uživatel tak vždy vidí jak model dronu, tak level s checkpointy, po kterém se pohybuje.

Při implementaci kamery jsem využil rozšíření Unity Cinemachine [51], které se specializuje na kameru pro hry a filmy. Výsledná kamera tak dynamicky reaguje na pohyb dronu. Při pohybu dronu doleva se rovněž natočí více doleva, při stoupání dronu se natočí směrem nad dron apod. Dron tak není vždy v úplném středu obrazovky. Díky tomu uživatel lépe vidí tím směrem, kterým se zrovna pohybuje. Také to působí, že dron po scéně téměř klouže, jelikož kamera dokončí pohyb oproti dronu s lehkým zpožděním a pohyb tak působí plynuleji, a to především u ovládání hand-trackingem, jelikož se rychlost pohybu nemění skokově, jako v případě ovládání klávesnicí.

Jak jsem již zmínil, pro ovládání hand-trackingem je potřeba zafixovat výchozí pozici dlaně. Lze tak učinit v menu, které uživatel vidí po spuštění aplikace, nebo po stisknutí klávesy Escape. Rovněž je v menu možné vybrat si preferovanou ruku. Pokud uživatel vůbec nezvolí výchozí pozici ruky, nebo ji zvolí příliš blízko snímatelnému rozsahu, UI na to uživatele upozorní a vyzve



■ **Obrázek 3.3** Uživatelské rozhraní aplikace v menu

ho k opravě. Pokud tak uživatel neučiní, nelze využít ovládání hand-trackingem. Může však stále využít ovládání klávesnicí. Pokud je výchozí pozice nastavena správně, v menu se text změní na „SET OK“ a hláška „keyboard only“ zmizí.

Po kliknutí na tlačítko START GAME, respektive CONTINUE, pokud hru již uživatel spustil, se spustí třívteřinový odpočet. Během těchto tří vteřin má uživatel čas umístit dlaň do výchozí pozice a připravit se na samotný let. Po jejich uplynutí se znovu spustí ovládání dronu, které je v menu pozastaveno, stejně jako časomíra.

### 3.5 Problémy a omezení

*Při implementaci jsem narazil mimo jiné na problém s kolizemi dronu a objektů ve scéně. Kolize ovlivnila dráhu letu i po té, co se dron objektu již nedotýkal. Problém se podařilo částečně vyřešit. Při spuštění aplikace na méně výkonných strojích pak může docházet k náhlým poklesům FPS, což negativně ovlivňuje ovládání.*

Vývoj aplikace v Unity přináší několik výhod a v mnoha aspektech ho usnadňuje. Lze tak využít například interní fyzikální model Unity, který mimo jiné řeší i kolize objektů. Aby engine kolizi zaznamenal, musí být součástí obou objektů tzv. Collider [52]. Ten slouží jako jakási slupka modelu, která detekuje kolize. Collider může mít zjednodušený tvar, či se může jednat o složitější, konvexní obal modelu. Pokud se dva či více objektů s neproniknutelnými Colliderly srazí, dojde na základě jejich pohybu, rychlosti, váze a tvaru k odražení, případně k deformaci – to zajišťuje třída Rigidbody [53].

V případě této práce je samozřejmě žádoucí, aby dron nemohl proletět skrze překážky ve scéně – ať už to jsou budovy, stromy, či jiné objekty. Při dlčím testování jsem však narazil na následující problém. Pokud dron kolidoval s některým objektem ve scéně, ovlivnilo to jeho dráhu letu i poté, co se ho již nedotýkal. Dron tak kupříkladu samovolně stoupal a rotoval, aniž by uživatel jakkoliv hýbal rukou.

Bylo to zřejmě způsobené tím, že pohyb dronu je implementován změnou atributů třídy Transform. Skript zajišťující pohyb dronu tak například chtěl dron posunout směrem vpřed, kam

ho ale nepustila srážka s překážkou ve scéně. Zřejmě tak docházelo ke kolizi fyzikálního modelu Unity, který chtěl, aby se dron od překážky odrazil, a skriptu MoveForward, který naopak chtěl, aby dron pokračoval dál vpřed.

Problém se mi částečně podařilo vyřešit následujícím způsobem. Zmíněná třída RigidBody umožňuje uzamknout pozici a rotaci objektu, aby nebyla ovlivněna v případě kolize. Zjistil jsem, že uzamčení a následné odemčení těchto atributů vynuluje nežádoucí efekt vzniklý kolizí dronu s překážkou. Ve skriptu CollisionFix tak vždy po ukončení kolize během jednoho snímku takto resetuji tyto atributy.

Stává se však, že příležitostně dron proletí skrze překážku ve scéně, i když mají oba objekty neproniknutelný collider. Může to způsobovat právě skript CollisionFix, nebo je to jen chyba Unity, každopádně se jedná o známé omezení, a je tak potřeba ho zde zmínit.

Další omezení souvisí se senzorem a hand-trackingem jako takovým. Při detekci rukou předává senzor do Unity každý snímek spoustu dat, kde dochází k jejich zpracování. Na méně výkonných zařízeních tak může docházet k poklesu FPS, což samozřejmě negativně ovlivňuje ovládání. Při malém výkonu pak může mít problém i samotný senzor, který pak ruku nerozpozná i přes to, že je v jeho snímatelném rozsahu.

Zmíněné komplikace částečně ovlivnily i následné uživatelské testování. Další problémy a omezení, které vyšly na povrch a případnou reakcí na ně, popisuji právě v kapitole 4 Testování .

### 3.6 Možnosti dalšího rozšíření

*Aplikaci by bylo možné rozšířit o další levely, případně vytvořit procedurální generátor levelů. Prospěl by jí i ukazatel high score jednotlivých levelů, či optimalizace ovládání pomocí klávesnice, případně ovladače. Nabízí se i myšlenka aplikovat navržené ovládání hand-trackingem na reálný dron.*

Pro účely této práce jsem vytvořil jeden výukový level a jeden ukázkový level s checkpointy. Tyto jednotlivé levely jsou samostatné spustitelné soubory. V budoucnu by bylo možné vytvořit další levely a navrhnout UI, přes které by si uživatel vybral, který level chce zrovna spustit. Zároveň by u se každého levelu hodila tabulka nejlepších časů, což by hráče motivovalo ke znovuhratelnosti.

Levely lze poměrně jednoduše vytvářet ručně. Stačí využít prefab DP\_Essentials ze zdrojového Unity projektu, vložit ho do 3D scény a rozmístit checkpointy – proces blíže popisuji v příloze B, případně ve videu [54]. V budoucnu by bylo možné navrhnout i procedurální generátor, který by mohl proces vytváření automatizovat.

Ovládání klávesnicí bylo pro tuto práci druhotné a v budoucnu by tak mohlo být více optimalizováno, či alespoň rozšířeno o pohyb do stran, který zde v současné době chybí. Případně by mohlo být přidáno i ovládání pomocí klasického, konzolového ovladače, které by zkušenějším pilotům dronů jistě vyhovovalo více, jelikož jsou na ovládání joysticky zvyklí více než na klávesnici. U ovládání hand-trackingem by mohla být přidána možnost změny citlivosti.

Dynamická kamera a použitý model dronu [49] sice napomáhají simulaci reálného pohybu, ale realističnosti by pomohly ještě další animace. Například náklon dronu směrem, který se pohybuje, či interpolace pohybu, aby dron ještě více klouzal po scéně.

Nakonec se nabízí myšlenka využít navržené ovládání i pro reálný dron. Tento krok by samozřejmě vyžadoval napojení na API dronu a úpravu skriptů, které se v současné době starají o pohyb virtuálního dronu v Unity. Nejedná se však o nereálnou vizi.



## Kapitola 4

# Testování

*V rámci uživatelského testování si celkem 6 testerů vyzkoušelo výslednou aplikaci. Nejprve prošli tutorial scénou a poté si zkusili i průlet levelem s checkpointy. Většina testerů se shodla, že jim ovládání pomocí hand-trackingu připadá intuitivní a imerzivní. Následně jsem adresoval připomínky, které vzešly z dotazníku a opravil některé chyby, které při testování vyšly najevo.*

### 4.1 Průběh testování

*Testování probíhalo vždy v oddělené místnosti, kde byl již připravený notebook se zapojeným senzorem. Tester se nejprve seznámil s aplikací a její ovládáním pomocí tutorialu. Poté si vyzkoušel i level, kde se snažil za co nejkratší čas proletět všemi checkpointy. Nakonec vyplnil dotazník, ve kterém zhodnotil celou aplikaci, především pak její ovládání.*

Uživatelského testování se zúčastnilo celkem 6 testerů. Všichni z nich byli muži v rozmezí 15–29 let. Čtyři z nich měli zkušenost s ovládáním reálného drona, nebo drona dokonce vlastní. Zbylí dva pak reálný ani virtuální dron nikdy neovládali. Téměř všichni testeři pak měli alespoň minimální zkušenost s aplikacemi, které využívají k ovládání sledování části těla uživatele – konkrétně Xbox Kinect, PlayStation Move, Oculus Quest a další.

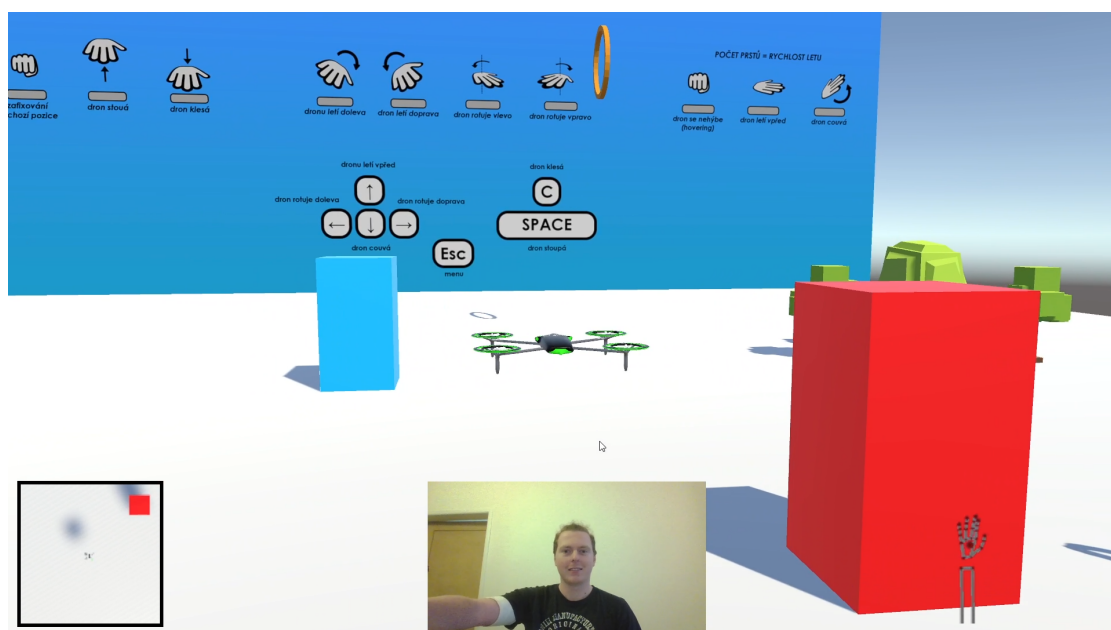
Testování probíhalo vždy jednotlivě v oddělené místnosti, kde byl na stole připraven notebook se zapojeným senzorem. Jednalo se o notebook Lenovo Legion 5 s následující konfigurací:

- CPU – AMD Ryzen 7 5800H (3,20 GHz)
- GPU – NVIDIA GeForce RTX 3060 Laptop
- RAM – 16 GB

Uživateli byl vždy představen koncept celé práce a princip ovládání. Následně si prošel tutorial, během kterého se seznámil s ovládáním pomocí hand-trackingu i pomocí klávesnice a především si obě tyto varianty vyzkoušel v praxi. Rovněž byl obeznámen s uživatelským rozhraním aplikace a systémem checkpointů.

Když si ovládání dostatečně osvojil, mohl se přesunout na průlet scénou s checkpointy. Zde si mohl lépe vyzkoušet přesnost ovládání. Opět přišel na řadu jak hand-tracking, tak klávesnice, přičemž si mohl ovládání vyzkoušet jak v sedě, tak ve stoje.

Během celého tohoto procesu byla nahrávána obrazovka počítače, záznam z kamery i zvukový záznam pomocí softwaru OBS Studio [55]. Kromě testera jsem byl v místnosti ještě já jakožto autor aplikace a moderátor celého testování. V průběhu jsem si zapisoval veškeré připomínky, pozitivní a především pak negativní feedback testerů.



■ **Obrázek 4.1** Jeden z testerů při průletu výukovou scénou

Testeři měli možnost strávit létáním libovolně dlouho. Většinou celé testování trvalo 30–50 minut a to včetně vyplnění závěrečného dotazníku. Ten měl tři části: v první zjišťoval, jaké má tester zkušenosti s drony a hand-trackingem, v druhé části vyžadoval porovnání ovládání hand-trackingem a klávesnicí spolu s výčtem jejich pozitiv a negativ, a v třetí části proběhlo zhodnocení UI a aplikace jako takové. Celý dotazník včetně jeho výsledků naleznete v příloze C.

## 4.2 Výsledky testování

*Uživatelské testování dopadlo uspokojivě. Výpovědi většiny testerů naznačují, že jim ovládání hand-trackingem přijde intuitivní a je vhodné pro ovládání dronu. Navíc by ho upřednostnili před ovládáním klávesnicí. V rámci UI se ukázala minimapa jako spíše zbytečná. Celkové hodnocení aplikace bylo velmi kladné. Nejčastější výtkou byl nedostatek levelů.*

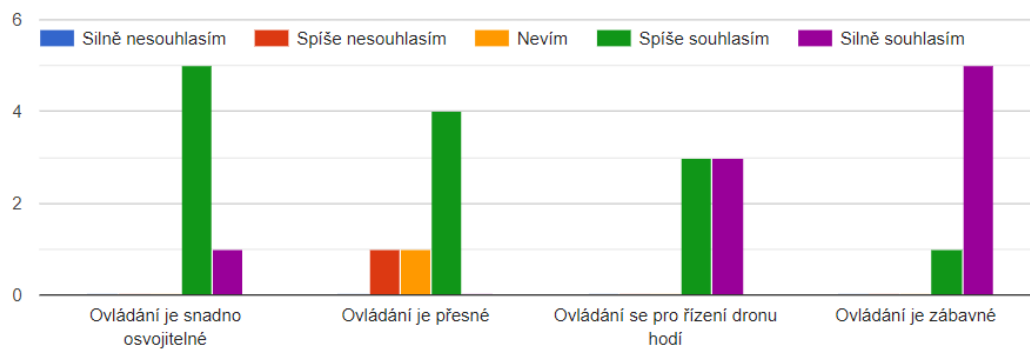
### 4.2.1 Ovládání

Začnu tím nejpodstatnějším, tedy ovládáním pomocí hand-trackingu. Zde se testeři shodli, že ovládání je lehce osvojitelné, pro řízení dronu se hodí a je velmi imerzivní. Co se týče přesnosti, měli dva z šesti lehké pochybnosti. Celkové hodnocení shrnuje graf 4.2.

V otevřených odpovědích pak testeři pochvalují právě imerzivitě ovládání, povětšinou dobře nastavenou citlivost a plynulost ovládání. Naopak vyčítají, že ovládání nemusí být vždy 100 % spolehlivé a může způsobit náhlé poklesy FPS aplikace. Dva z testerů pak měli ze začátku lehký problém s ovládáním výšky – přišlo by jim intuitivnější měnit výšku letu náklonem dlaně nahoru a dolů. Zbývá čtyři neměli k vertikálnímu pohybu výtky. Někteří testeři si také stěžovali, že je při delším používání bolí rameno.

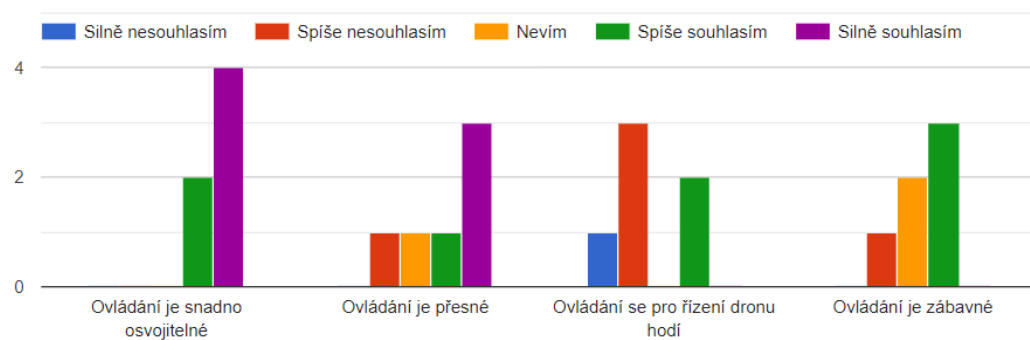
Ovládání pomocí klávesnice se pak ukázalo jako o něco přesnější. Všichni testeři se shodli, že je velice snadno osvojitelné, jelikož odpovídá běžnému arkádovému ovládání většiny závodních her. Zároveň ale převládá názor, že se klávesnice pro ovládání dronu spíše nehodí, a že ovládání je méně imerzivní. Výsledky shrnuje graf 4.3.

Jak hodnotíte ovládání dronu pomocí **hand trackingu**? U každého výroku vyberte, do jaké míry s ním souhlasíte:



**Obrázek 4.2** Výsledky hodnocení ovládání hand-trackingem

Jak hodnotíte ovládání dronu pomocí **klávesnice**? U každého výroku vyberte, do jaké míry s ním souhlasíte:



**Obrázek 4.3** Výsledky hodnocení ovládání klávesnicí

	Ovládání je snadno osvojitelné	Ovládání je přesné	Ovládání se pro řízení dronu hodí	Ovládání je zábavné
hand tracking	7	3	9	11
klávesnice	10	6	-1	2

■ **Tabulka 4.1** Výsledky srovnání ovládání hand-trackingem a klávesnicí

Největší výtkou pak byla v porovnání s hand-trackingem malá plynulost a menší svoboda pohybu. Celkově se všichni testeré shodli, že kdyby aplikaci použili znovu, volili by raději ovládání hand-trackingem – právě kvůli plynulosti, intuitivnosti a imerzivitě. Obě varianty ovládání jsem ještě porovnal v tabulce 4.1. Jednotlivým odpovědím jsem přiřadil následující bodové ohodnocení:

- Silně nesouhlasím (-2 body)
- Spíše nesouhlasím (-1 bod)
- Nevím (0 bodů)
- Spíše souhlasím (+1 bod)
- Silně souhlasím (+2 body)

Číslo v každé buňce je pak součtem všech odpovědí pro danou kategorii. Jelikož bylo celkem šest testerů, minimální možný a tím i nejhorší výsledek je -12 bodů, naopak nejlepší možný je 12 bodů. Jednotlivé buňky jsou barevně odlišeny dle výsledku, kde sytě červená odpovídá nejnižším číslům a sytě zelená odpovídá nejvyšším číslům.

Z tabulky 4.1 vidíme, že největší rozdíl byl v posledních dvou aspektech ovládání. Zároveň se zde potvrzuje, že ovládání klávesnicí je snadnější a přesnější, ale hand-tracking je se pro ovládání hodí výrazně více a je mnohonásobně zábavnější a imerzivnější. Částečně to může být způsobeno tím, že na ovládání klávesnicí nebyl v rámci této práce kladen takový důraz.

## 4.2.2 UI a celkové hodnocení

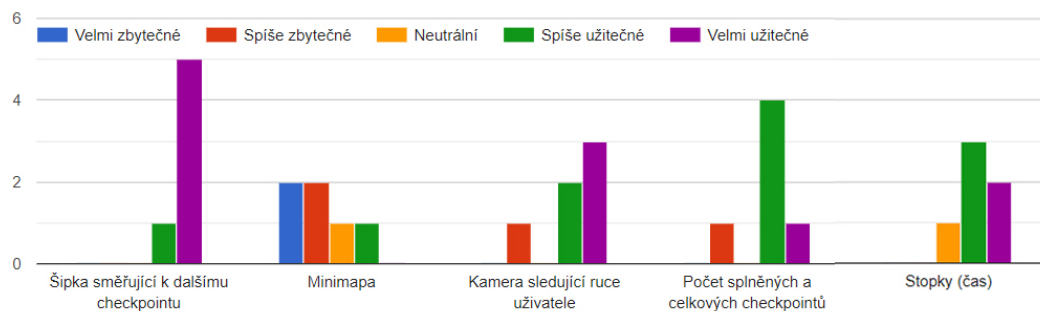
V poslední části dotazníku hodnotili uživatelé jednotlivé prvky uživatelského rozhraní a jejich přínos, a aplikaci jako takovou. Nejužitečnější se ukázala šipka ukazující následující checkpoint, nejméně užitečnou pak minimapa. Žádný z testerů však neuvedl, že by mu některý z prvků UI překážel, nebo vadil. Celkový přínos jednotlivých prvků shrnuje graf 4.4.

Obdobně jako u ovládání, i zde jsem jednotlivé odpovědi bodově ohodnotil, jejich součet vynesl do tabulky 4.2, a bodový zisk barevně oddělil. Zde je ještě více zřejmé, že orientaci ve scéně zastává primárně šipka ukazující další checkpoint, oproti tomu minimapa pouze minimálně. Zbylé prvky UI se zdají být vcelku prospěšné.

V otevřených odpovědích pak bylo několik poznámek k uživatelskému rozhraní menu. Testerům chybělo tlačítko pro restart levelu a to jak po dokončení levelu, tak i v průběžném pause menu. Dále by ocenili indikátor, že nově zafixovaná výchozí pozice v menu je nejen ve validních mezích, ale že se po sevření pěsti skutečně aktualizovala. Nakonec zde padla myšlenka přidat vizualizaci výchozí pozice ruky, která by uživateli napověděla, zda je nad ní a tedy stoupá, nebo naopak.

Dále hodnotili testeré aplikaci jako celek. Chválili především ovládání hand-trackingem, zábavnost, kompetitivnost a prostředí levelů. Také vyslovili přání, aby byla hra rozšířena o další levely. Rovněž by si přáli přidat ke každému levelu tabulku nejrychlejších časů – scoreboard.

Který z prvků uživatelského rozhraní Vám přišel užitečný, či naopak zbytečný?  
 Vyberete vždy jednu možnost pro každý prvek:



**Obrázek 4.4** Výsledky hodnocení uživatelského rozhraní

	šipka směřující k dalšímu checkpointu	minimapa	kamera sledující ruce uživatele	počet splněných a celkových checkpointů	stopky (čas)
UI	11	-5	7	5	7

**Tabulka 4.2** Výsledky srovnání užitečnosti prvků UI

U dvou testerů pak došlo na známý problém s kolizemi, který jsem popsal již v implementaci. Rovněž byl v některých případech problém s nízkými FPS, čemuž nahrávání systémem OBS příliš nepomohlo. V takových případech jsem proto nechal testery vyzkoušet si aplikaci i „mimo záznam“, kde již aplikace běžela o něco plynuleji. Další výtky a poznámky, a především případnou reakci na ně, rozebírám podrobněji v následující kapitole 4.3.

### 4.3 Reakce na testování

*V této kapitole adresuji jednotlivé připomínky a výtky, které vzešly z uživatelského testování. U každé popisuji, v čem spočívala, jak byla vyřešena, případně proč řešena nebyla. Jako nejzásadnější se zdálo přidat tlačítko pro restart levelu a potvrzení o aktualizované výchozí pozici ruky. Také byl přidán další level a jednoduchý ukazatel nejrychlejšího času, který by však v budoucnu zasloužil vylepšit.*

#### Nízké FPS a neplynulý běh aplikace

Výpadky FPS byly částečně způsobeny systémem OBS, který při nahrávání obrazovky, kamery a zvuku tak ještě více zatěžoval počítač, na kterém aplikace běžela. Po vypnutí nahrávání se situace zlepšila. To nic nemění na faktu, že pro hand-tracking je potřeba každý snímek přenést nemalé množství dat, a pro dosažení nejlepších výsledků je tak vhodné spouštět aplikaci na co nejvýkonnějším zařízení.

#### Nedostatečně optimalizované ovládání klávesnicí

Ovládání klávesnicí je pro tuto práci druhotné a slouží pouze jako alternativa k hand-trackingu. Tento problém tak nebyl dále řešen a může být předmětem dalších rozšíření.

#### Bolest ramena při dlouhodobějším ovládání hand-trackingem

Zde částečně záleží, jak vysoko si uživatel nastaví výchozí pozici pro ovládání výšky letu. Také může pomoci při ovládání stát, případně si dělat krátké pauzy – let lze kdykoliv pozastavit klávesou Escape. Jinak se ale jedná o riziko, které sebou ovládání rukou nese, podobně jako například bolavé oči u používání VR headsetu.

#### Tendence ovládat dron jinými gesty

Dva z testerů měli tendenci ovládat pohyb nahoru/dolů náklonem dlaně. Jak zmiňuji v implementaci, snímání prstů v pozici, kdy jsou téměř kolmé na senzor, může být problémové. Kdyby se však rozsah náklonu nastavil správně, mohlo by se jednat o alternativní způsob ovládání, kde by navíc odpadla potřeba zafixovat výchozí pozici ruky. Tuto možnost by mohla prozkoumat další práce, která by navíc mohla porovnat tyto dvě varianty ovládání výšky letu.

#### Změna citlivosti ovládání

Při otázce na citlivost ovládání odvětila většina z dotázaných testerů, že by ji neměnila. Pouze jeden z nich uvedl, že do budoucna by možnost změny citlivosti uvítal. V současné době je takto nastavené ovládání dostačující a úprava citlivosti by tak mohla být předmětem budoucího rozšíření aplikace.

#### Chybějící potvrzení o aktualizaci výchozí pozice ruky

Při změně výchozí pozice v menu chyběl jakýkoliv feedback, že byla pozice skutečně aktualizována. Pod ukazatel, zda je výchozí pozice nastavena správně, proto přibyl indikátor, v jaké



■ **Obrázek 4.5** Aktualizované UI menu

konkrétní výšce (v centimetrech) je výchozí pozice nastavena. Změnu pozice tak uživatel vidí na změně tohoto čísla. Zároveň má díky tomu uživatel lepší představu, v jaké výšce je nulová hladina, ve které dron nestoupá ani neklesá.

### Chybějící tlačítko pro restart levelu

Na tento problém si stěžovali snad všichni z šesti oslovených testerů. Zdálo se tedy téměř imperativní ho opravit. Po dokončení levelu tak má nyní uživatel možnost hru buď ukončit, či restartovat. Restart vrátí dron do původní pozice, vynuluje časomíru i počet splněných checkpointů. Tlačítko pro restart přibylo i do Escape menu.

### Minimální využití minimapy

Testování ukázalo, že minimapa není příliš využívaným prvkem uživatelského rozhraní. Je to částečně tím, že orientaci ve scéně dostatečně zastává šipka ukazující následující checkpoint. Nikomu z testerů však minimapa nevadila, ani nepřekážela. Proto jsem se rozhodl ji v UI ponechat. V budoucnu by mohla být doplněna o ikony a ukazatele, které by jí přidaly na užitečnosti.

### Chybějící high score, či scoreboard

Většina testerů se shodla, že přidání high score, tedy doposud nejrychlejšího času, by aplikaci pomohlo a zvýšilo by to kompetitivnost. Do UI jsem tak z časových důvodů přidal alespoň „jednorázový“ ukazatel nejlepšího času, který má však platnost pouze po dobu běhu programu. Po ukončení aplikace se bohužel vynuluje. V budoucnu by bylo dobré, aby se tento čas lokálně ukládal, případně by se mohla přidat celá tabulka nejlepších časů, do které by se výsledky ukládaly.

### Problém s kolizemi a hranicemi levelu

Poloprůhledná krychle, která objímá celý ukázkový level by měla zamezit, aby se dron dostal mimo hranice levelu. Při dostatečné míře vytrvalosti se však podařilo dvěma testerům z tohoto prostoru vylézt. Částečně pomohlo krychli nahradit průhlednými deskami, avšak i zde se příležitostně projeví známý problém s kolizemi a dron projde skrz. Nabízí se tak možná raději scénu neuzavírat a spoléhat na šipku ukazující správný směr a v případě, že by uživatel odletěl daleko mimo scénu level ručně restartovat.

### Více animací při pohybu dronu

Konkrétně například náklon dronu ve směru, kterým se pohybuje, či větší setrvačnost pohybu. Tyto funkcionality však nebyly dále řešeny a mohou být předmětem dalších rozšíření.

### Přidání dalších levelů

Také tento požadavek se objevil několikrát. Vytvořil jsem proto ještě jeden level s celkem 16 checkpointy. Jako model jsem využil demo scénu balíčku Prototype Pack [56]. Tento level je o něco náročnější, jelikož vyžaduje přesné průlety skrz úzké otvory, rychlejší změny směru apod. Vytváření dalších levelů pak do budoucna usnadní vytvořené prefabs v přiloženém Unity projektu.



## Závěr

Cílem této diplomové práce bylo vytvořit jednoduchou aplikaci zaměřenou na ovládání virtuálního dronu pomocí hand-trackingu a ověřit, zda je tento způsob pro ovládání dronu vhodný.

V práci jsem se nejprve zabýval analýzou existujících řešení. Ta mi poskytla vhled do problematiky hand-trackingu, přiblížila jednotlivé možnosti a technologie. Také jsem v rámci ní prozkoumal současná řešení. Dále jsem se zaměřil na drony jako takové, s důrazem na jejich ovládání. Blíže jsem se také seznámil se senzory OAK-D-Lite a Leap Motion Controller od společnosti Ultraleap. Nakonec jsem prozkoumal možnosti a úskalí vytvoření aplikace s podporou hand-trackingu. Výsledek analýzy byl takový, že téma práce je velice aktuální a má tak smysl se jím zabývat.

Ve fázi návrhu aplikace padlo rozhodnutí využít pro hand-tracking senzor Ultraleap Stereo IR 170 a pro vývoj software Unity. O komunikaci mezi senzorem a aplikací se postaral Ultraleap Plugin for Unity. Nejzásadnější byl pak samozřejmě samotný návrh ovládání dronu pomocí hand-trackingu. Ten vypadal následovně: pohyb ve vodorovné rovině pomocí náklonu dlaně, rotace dronu do stran pomocí rotace dlaně a klesání/stoupání dronu pomocí vertikálního pohybu dlaně. Dále jsem navrhl jednoduché UI aplikace a rozebral myšlenku gamifikace celé aplikace a s tím spojenou strukturu scén, po kterých by se měl virtuální dron pohybovat.

Při implementaci pak tyto myšlenky nabraly konkrétních tvarů. Ovládání prošlo od návrhu několika změnami, tou nejzásadnější bylo navázání rychlosti pohybu na počet zdvižených prstů, nikoliv na míru náklonu dlaně vpřed. Pro ovládání výšky letu je potřeba gestem pěti zafixovat výchozí pozici dlaně. Pokud je dlaň nad touto rovinou, dron stoupá – obdobně pro klesání. Kromě ovládání hand-trackingem jsem naimplementoval i jednoduché ovládání pomocí klávesnice. Výslednou aplikaci jsem proměnil v jednoduchou hru, kde je uživatelským cílem proletět všemi checkpointy ve scéně za co nejkratší čas. Vytvořil jsem jednu výukovou scénu, kde se uživatel seznámí se specifickým ovládáním a další dvě scény, kde nabyté zkušenosti aplikuje v praxi.

Poslední fází bylo uživatelské testování, které prověřilo především implementované ovládání pohybem ruky, ale i ovládání klávesnicí, uživatelské rozhraní a fungování aplikace jako takové. Testeři nejprve prošli tutorial scénou a poté si zkusili i průlet levelem s checkpointy. Nakonec vyplnili dotazník, ve kterém celý průběh testování zhodnotili. Zbývalo adresovat připomínky, které vzešly z dotazníku a opravit některé chyby, které při testování vyšly najevo.

Ukázalo se, že výsledné ovládání virtuálního dronu je dostatečně intuitivní, imerzivní a vcelku snadno osvojitelné. Všichni z testerů uvedli, že by ho upřednostnili před ovládáním klávesnicí. Bylo tak dosaženo hlavního cíle této práce – ovládání rukou je navíc dostatečně robustní a poměrně přesné. Rovněž gamifikace celé aplikaci prospěla.

Práce tak položila solidní základ pro implementaci alternativního ovládání reálného dronu. V budoucnu by mohla být aplikace ještě rozšířena o další levely, u kterých by si udržovala tabulku nejrychlejších časů – to by zvýšilo kompetitivnost. Rovněž by prospělo, aby měl pohyb dronu

větší setrvačnost a naklonil se vždy do směru pohybu – tím by se pohyb ještě přiblížil realitě. Nakonec by mohlo dojít k optimalizaci ovládní klávesnicí, které bylo pro tuto práci druhotné.

## Slovník pojmů

- **Inside-out tracking** – způsob sledování polohy uživatele, kdy se senzor hýbe spolu s uživatelem a skenuje okolí [57]
- **Outside-in tracking** – způsob sledování polohy uživatele, kdy nehybné senzory rozmístěné po scéně skenují (nejen) uživatele [57]
- **Thresholding** – zmenšení, či zvětšení hodnoty, aby byla v určitých mezích (prahování)
- **Unity prefabs** – přednastavené a optimalizované objekty připravené k použití v Unity



## Spuštění aplikace

Před samotným spuštěním aplikace se přesvědčte, že máte na svém zařízení nainstalovanou aplikaci Ultraleap Tracking [58]. Bez ní ovládání pomocí hand-trackingu nelze využít a budete tak odkázáni pouze na klávesnici. Po instalaci se můžete přesvědčit, že hand-tracking probíhá správně. Stačí zapojit senzor a otevřít kontrolní panel v aplikaci Ultraleap.

Nyní je vše připraveno pro spuštění výsledné aplikace, která vznikla v rámci této diplomové práce. Umístěte senzor na stůl tak, abyste nebyli nijak omezeni v pohybu a rotaci ruky. U ovládání můžete sedět, či stát – záleží na vaší preferenci. Myslete však na částečně omezený rozsah snímání senzoru.

Otevřete jeden ze spustitelných souborů, které naleznete na přiloženém CD, nebo na GitLabu [59]. Doporučuji začít TUTORIAL scénou, seznámit se s ovládáním a až následně se pustit do scény LEVEL\_01. Pokud dáváte přednost video materiálům před psaným textem, můžete shlédnout krátké instruktážní video [54], ve kterém popisují, jak aplikaci nastavit a samotné ovládání hand-trackingem.

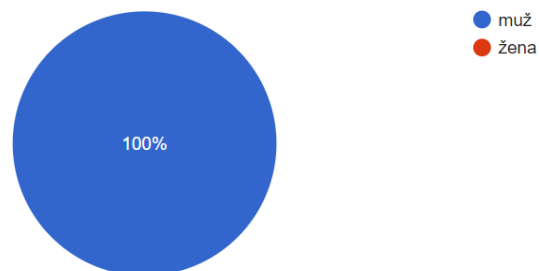
Až vás omrzí přiložené scény, můžete si vytvořit vlastní pomocí Unity prefabs v přiloženém Unity projektu. Stačí vložit prefab „DP Essentials“ do vámi vytvořené scény a udělat z něj samostatnou, lokální kopii pomocí tlačítka „Unpack“. Poté stačí rozkopírovat po scéně prefab Checkpoint a v objektu MyGameManager nastavit jejich pořadí. Tento proces ručního vytváření dalších levelů rovněž popisují ve zmíněném videu [54].



# Kompletní výsledky testování

Pohlaví:  
6 responses

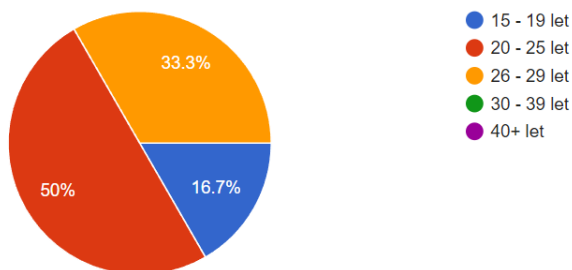
 Copy



Věk:

 Copy

6 responses



Máte zkušenost s ovládáním reálného, či virtuálního dronu?

 Copy

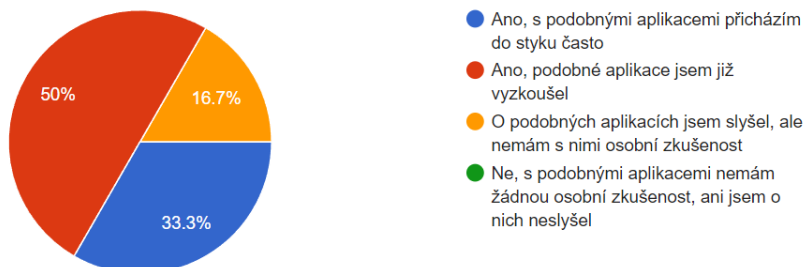
6 responses



Máte zkušenost s aplikacemi, které využívají k ovládání sledování části těla uživatele? (hand tracking, Kinect, apod.)

 Copy

6 responses





Pokud ano, vyjmenujte konkrétní aplikace, či technologie, se kterými máte osobní zkušenost:

5 responses

Xbox Kinect, Oculus Quest

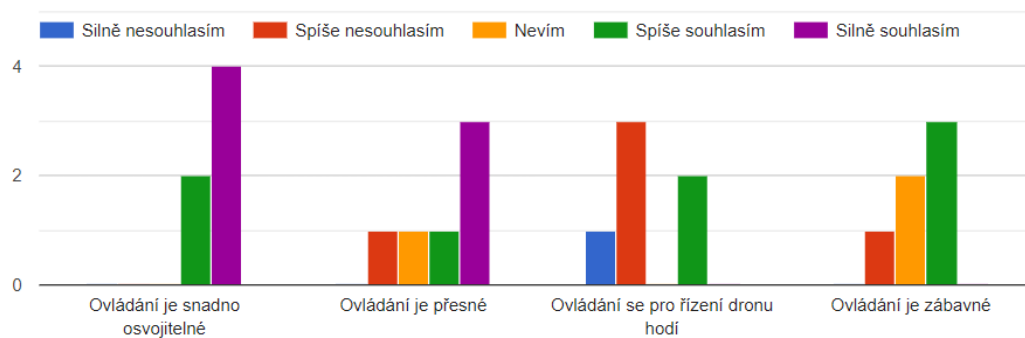
V autě snímání gest ruky

Playstation move, Vyzkoušel jsem leap motion a kinect

PS3 Kinect - Just dance, Eye tracking pomocí webkamery, Oculus quest 2 - Hand spell,...

Oculus Quest 2  
Kinect  
PS VR

Jak hodnotíte ovládání dronu pomocí **klávesnice**? U každého výroku vyberte, do jaké míry s ním souhlasíte:



Co a proč se Vám na ovládání pomocí klávesnice **líbilo**?

6 responses

Každý s alespoň malou zkušeností s pc hrami dokáže ovládání rychle pochopit.

Je poměrně jednoduché, možná až moc, ale člověk se ho alespoň naučí téměř okamžitě...

Je to hodně intuitivní, pro lidi co někdy hráli na počítači

Ovládání aplikace pomocí klávesnice bylo přesné a nedělalo žádné problémy

Okamžitá a přesná odezva, okamžité osvojení kvůli podobnosti s ostatními hrami

Odpovídá to běžnému arkádovému ovládání závodních her.

Co a proč se Vám na ovládání pomocí klávesnice **nelíbilo**?

6 responses

ovládání není přesně a pro ovládání dronu se nehodí. Nejde nastavovat intenzita zatáčení, rychlost atd.

Ovládání není moc plynulé, spíš takové trhané. Neumožňuje takovou svobodu pohybu. Možná by bylo lepší využít i myš pro určení směru/rotaci dronu, nebo více kláves

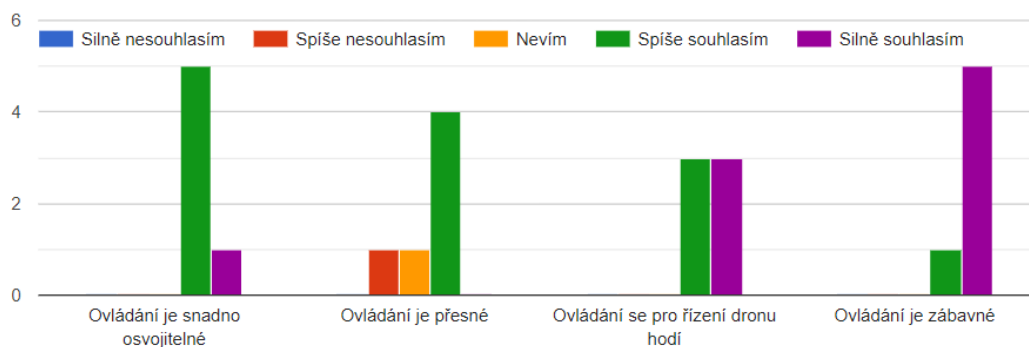
Nebylo moc plynulé

Neměl jsem takovou svobodu ovládání jako pomocí ruky a bylo méně zábavné

Chybějící možnost náklonu, celkově menší zábava

Absence jakéhokoliv působení fyziky na model. Setrvačnost mi v případě ovládání šipkami chyběla.

Jak hodnotíte ovládání dronu pomocí **hand trackingu**? U každého výroku vyberte, do jaké míry s ním souhlasíte:



### Co a proč se Vám na ovládání pomocí hand trackingu **líbilo**?

6 responses

Intuitivnost a ovládání dronu pouze pohybem ruky v prostoru. (je to magie)

Ovládání je intuitivnější, než jsem čekal. Podle tutoriálu působí poněkud složitě, ale když si ho člověk vyzkouší, je to poměrně přirozené. Oproti klávesnici má více možností - různá rychlost letu, pohyb do stran, plynulejší otáčení apod.

Gesta ruky: ovládání pomocí gest je snadno naučitelné, je v nich zahrnuto vše co k řízení dronu potřebuji. Když si člověk zvykne na rádius snímání, tak to funguje skvěle. Za mě je nastavená dobrá citlivost na ovládání dronu, jen u couvání by potřebovala zmírnit, někdy se stane že dron začne couvat i když nechci.

Ovládání bylo zábavné a imerzivní (ovládání bylo přirozené a dokážu si představit, že bych ho použil i při ovládání normálního dronu --> mezi rizika však patří bolest ruky po nadměrném užívání, kolize dronu z důvodu nepřesnosti ovládání, či umístění leap motinu v terénu), také se mi líbilo rozpoznání preferované ruky a možnost používat druhou ruku k jiným činnostem

Více se hodí pro ovládání dronu, můžete se pomocí toho téměř vtělit do samotného dronu

Bylo to něco nového, zaujala responzivita zařízení a dokázal bych si představit, že bych to zkusil znovu.

### Co a proč se Vám na ovládání pomocí hand trackingu **nelíbilo**?

6 responses

senzor musí být na správném místě a odpojí se jakmile se ruka příliš vychýlí z jeho zorného pole.

Trochu jsem narážel na spolehlivost senzoru, který občas ruku "neviděl". Možná by pomohlo i nastavení citlivosti ovládání

Musí si člověk zvyknout na rádius senzoru. Citlivost při couvání

- nějaké pohyby jsem měl tendenci provádět jiným gestem (například stoupání a klesání, jsem měl tendenci dělat náklonem ruky dopředu a dozadu),
- při couvání bylo znemožněno zatáčení,
- chyběla možnost kalibrace ovládání --> s rukou jsem musel vytáčet aby byla srovnána s leap motionem),
- Při překřížení rukou (pravá ruka je na levé straně a levá ruka je na pravé straně) začne leap motion snímat špatnou ruku"
- nutnost mít konstantně zvednutou ruku což po delším používání může vést k bolestem

Omezená možnost udělat si pauzu (ale pokud by šlo o použití v reálném světě, tak to přestává být negativum)

Nespolehlivost senzoru a neuplně vhodně zvolená gesta pro ovládání dronu. Nebylo to natolik intuitivní, jak létat například na leteckém simulátoru s joystickem.

Pokud byste aplikaci použil/a znovu, který způsob ovládání (klávesnice nebo hand tracking) byste zvolil/a a proč?

6 responses

hand tracking, je intuitivnější a pro ovládání dronu přirozenější než klávesnice.

hand tracking, je plynulejší, zábavnější a i přes lehce nižší spolehlivost se mi s ním dron ovládal lépe

Ovládání pomocí hand trackingu, hodí se to mnohem víc na ovládání dronu než pomocí klávesnice. Je to plynulé, u klávesnice při otáčení to tak plynulé není. Ovládání gesty je zábavné a rychle se to člověk naučí.

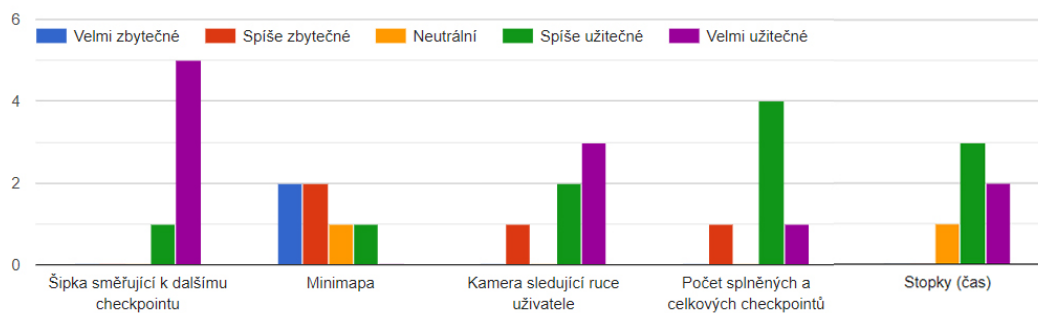
Hand tracking --> je to zábavnější, připadá mi přirozenější a má více možností, i když občas dochází k mírným nepřesnostem, také občas zapříčiňovalo sekání aplikace

Hand tracking - mnohem lepší pro vcítění se do samotného dronu

Pokud by byl víc odladěný hand tracking, zvolil bych hand tracking. Primárně z důvodu přesnosti pohybu, kterého nelze na klávesnici docílit.

Který z prvků uživatelského rozhraní Vám přišel užitečný, či naopak zbytečný?

Vyberete vždy jednu možnost pro každý prvek:



Je něco, co Vám v uživatelském rozhraní aplikace **chybělo**, či naopak **překáželo**?

6 responses

Ne.

Na minimapu jsem se v podstatě nepodíval a když ano, tak mi neřekla nic navíc. Nepřekážela mi tam, ale nepřijde mi, že by byla potřeba, jelikož šipka ukazující směr už člověka navede dobře

Při resetování nastavení hladiny ruky, jsem si nebyl jistý, jestli se to přenastavilo nebo ne.

Tlačítko restartu, či ukazatel nastavené neutrální výšky (hodil by se nějaký ukazatel, který by byl uprostřed když jsem v neutrální pozici, a stoupal by nahoru podle toho jak moc zvednu ruku, respektivě dolů podle toho jak moc klesnu s rukou --> hodilo by se i klepšímu pochopení ovládání, se kterým jsem měl menší problém)

Více levelů, možnost vypnout rotaci vrtulí pro rychlé klesání

Šipka ze začátku měla špatný kontrast s pozadím hry a netušil jsem kam mam vlastně letět. Hand tracking kamera je fajn ale při letu prakticky k ničemu. Spíše bych upřednostnil text či informaci že tracking je v pořádku či v nepořádku.

Co se Vám na aplikaci jako takové **líbilo** a co Vás nejvíce **zaujalo**?

6 responses

Působení uceleně. Má jasný cíl, jasné ovládání a díky stopkám hráč může sledovat své zlepšení.

Ovládání rukou je zábavné a funguje poměrně dobře. Tutoriál na začátku působí trochu krkolomně a vypadá trochu vyčerpávající, ale pak je ovládání rukou fajn, i když z něj po delší době trochu bolí ramena. Dron poměrně hezky "klouže" po scéně.

Aplikace je celkově velmi vydařená a zábavná. Líbilo se mi ovládání pomocí hand trackingu.

Možnost kompetitivnosti a zajímavé prostředí umožňující vyzkoušení všech prvků

Létání v reálném prostředí, přesnost ovládání

Její jednoduchost a potenciál na rozšíření.

Co se Vám na aplikaci jako takové **nelíbilo**, nebo Vám v ní **chybělo**?

6 responses

Možná více levelů.

Při jednom z průletů jsem se dostal mimo hranice levelu a chvíli jsem se nemohl dostat zpátky - přepokládám, že je to bug. Ovládání klávesnicí není nic moc. Když už je let na čas, tak bych možná přidal nějaké high score. Škoda, že je jen jeden level na čas, klidně bych si jich vyzkoušel více... Možná bych ještě více animoval dron, aby se nakláníl do směru letu apod. Ale aspoň vrtulky se točí :D

Asi možnost restartovat po dokončení kola.

Proletování modely a žádné ohraničení prostředí

Občasný pokles FPS, možnost zvýšení FOV

Absence setrvačnosti pro model dronu a naklápění modelu podle rotace ruky. Víím že to bylo nejspíše nad rámec této práce, ale přijde mi že by to zlepšilo celkový zážitek z letu.

Napadá Vás, jak aplikaci **vylepšit**, či dále **rozšířit**?

4 responses

Poladit ovládání klávesnicí, přidat další levely, high score, nastavení citlivosti

Score board, collidery, ohraničení mapy (abych nemohl vylézt pryč)

Více levelů, rychlejší drony

Úroveň kde by se uživatel potýkal např. se silným větrem nebo v dešti. Aby to mohl být "reálný" тренаžér. S tím se nejspíše pojí i to, co jsem vytkl v předešlé otázce, co mi na hře chybělo.

# Bibliografie

1. NOVACEK, Tomas; JIRINA, Marcel. Overview of Controllers of User Interface for Virtual Reality. *PRESENCE: Virtual and Augmented Reality*. 2020, roč. 29, s. 37–90. ISSN 1054-7460. Dostupné z DOI: 10.1162/pres\_a\_00356.
2. HTC CORPORATION. *About the VIVE base stations*. Microsoft Corporation, [b.r.]. Dostupné také z: [https://www.vive.com/hk/support/vive/category\\_howto/about-the-base-stations.html](https://www.vive.com/hk/support/vive/category_howto/about-the-base-stations.html).
3. NOVACEK, Tomas. *Technologie virtuální reality – Ovladače pro VR*. CTU, Faculty of Information Technology, 2022. Dostupné také z: [https://docs.google.com/presentation/d/18f61UFvaLVsa6fuQLsCK4EFdFkyVEqWvrIGT2TJcIGc/edit#slide=id.geff9dcf6eb\\_0\\_181](https://docs.google.com/presentation/d/18f61UFvaLVsa6fuQLsCK4EFdFkyVEqWvrIGT2TJcIGc/edit#slide=id.geff9dcf6eb_0_181).
4. SARBOLANDI, Hamed; LEFLOCH, Damien; KOLB, Andreas. Kinect range sensing: Structured-light versus Time-of-Flight Kinect. *Computer Vision and Image Understanding*. 2015, roč. 139, s. 1–20. ISSN 1077-3142. Dostupné z DOI: <https://doi.org/10.1016/j.cviu.2015.05.006>.
5. ULTRALEAP. *Ultraleap Hand Tracking: Technical Introduction*. 2020. Dostupné také z: <https://cms.ultraleap.com/app/uploads/2020/07/Ultraleap-Hand-Tracking-Technical-Introduction-1.pdf>.
6. NOVÁČEK, Tomáš; JIŘINA, Marcel. Project MultiLeap: Making Multiple Hand Tracking Sensors to Act Like One. In: *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. 2021. Dostupné z DOI: 10.1109/AIVR52153.2021.00021.
7. LUXONIS. *OAK-D-Lite Datasheet*. 2021. Dostupné také z: [https://github.com/luxonis/depthai-hardware/blob/master/DM9095\\_OAK-D-LITE\\_DepthAI\\_USB3C/Datasheet/OAK-D-Lite\\_Datasheet.pdf](https://github.com/luxonis/depthai-hardware/blob/master/DM9095_OAK-D-LITE_DepthAI_USB3C/Datasheet/OAK-D-Lite_Datasheet.pdf).
8. META PLATFORMS. *Getting started with Hand Tracking on Meta Quest headsets*. 2023. Dostupné také z: <https://www.meta.com/help/quest/articles/headsets-and-accessories/controllers-and-hand-tracking/hand-tracking/>.
9. VIVE TUTORIALS. *Using Hand Tracking in VIVE Focus 3*. 2022. Dostupné také z: <https://www.youtube.com/watch?v=DFth2IVfm5Ms>.
10. PIMAX. *Pimax Hand Tracking Module*. [B.r.]. Dostupné také z: <https://pimax.com/product/hand-tracking-module/>.
11. ULTRALEAP. *MOUNT CAMERA ON A VR HEADSET*. 2016. Dostupné také z: <https://cms.ultraleap.com/app/uploads/2020/07/Ultraleap-Hand-Tracking-Technical-Introduction-1.pdf>.

12. BARTÁK, Roman; VYKOVSKÝ, Adam. Any Object Tracking and Following by a Flying Drone. In: *2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI)*. 2015, s. 35–41. Dostupné z DOI: 10.1109/MICAI.2015.12.
13. KALAL, Zdenek. *Tracking Learning Detection*. Guildford: University of Surrey, 2011. Dis. pr. University of Surrey (United Kingdom).
14. SANNA, Andrea; LAMBERTI, Fabrizio; PARAVATI, Gianluca; MANURI, Federico. A Kinect-based natural interface for quadrotor control. *Entertainment Computing*. 2013, roč. 4, č. 3, s. 179–186. ISSN 1875-9521. Dostupné z DOI: <https://doi.org/10.1016/j.entcom.2013.01.001>.
15. MURTAZA'S WORKSHOP – ROBOTICS AND AI. *Hand Gesture Drone Control — OpenCV Python*. 2021. Dostupné také z: [https://www.youtube.com/watch?v=dU9S811D\\_Es](https://www.youtube.com/watch?v=dU9S811D_Es).
16. MACCHINI, Matteo; HAVY, Thomas; WEBER, Antoine; SCHIANO, Fabrizio; FLOREANO, Dario. Hand-worn Haptic Interface for Drone Teleoperation. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, s. 10212–10218. Dostupné z DOI: 10.1109/ICRA40945.2020.9196664.
17. ZHAO, Zhenfei; LUO, Hao; SONG, Guang-Hua; CHEN, Zhou; LU, Zhe-Ming; WU, Xiaofeng. Web-based interactive drone control using hand gesture. *Review of Scientific Instruments*. 2018, roč. 89, č. 1, s. 014707. ISSN 0034-6748. Dostupné z DOI: 10.1063/1.5004004.
18. SZ DJI TECHNOLOGY CO., LTD. *DJI Spark Specs*. [B.r.]. Dostupné také z: <https://www.dji.com/cz/spark/info#specs>.
19. FIREFLY DRONE. *Firefly Drone™ – The only drone in the world you control with your hand!™*. 2019. Dostupné také z: [https://www.youtube.com/watch?v=H-RJ2S\\_b0Jk](https://www.youtube.com/watch?v=H-RJ2S_b0Jk).
20. CAUCHARD, Jessica R.; E, Jane L.; ZHAI, Kevin Y.; LANDAY, James A. Drone & Me: An Exploration into Natural Human-Drone Interaction. In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Osaka, Japan: Association for Computing Machinery, 2015. UbiComp '15. ISBN 9781450335744. Dostupné z DOI: 10.1145/2750858.2805823.
21. ULTRALEAP. *Ultraleap mid-air haptics*. [B.r.]. Dostupné také z: <https://docs.ultraleap.com/haptics/#ultraleap-mid-air-haptics>.
22. ULTRALEAP. *LeapMotion-Feature*. [B.r.]. Dostupné také z: <https://mocap.reallusion.com/iclone-motion-live-mocap/leap-motion.html>.
23. LUXONIS. *OAK-D-Lite Overview*. [B.r.]. Dostupné také z: <https://docs.luxonis.com/projects/hardware/en/latest/pages/DM9095/#overview>.
24. LUXONIS. *Auto-Focus vs Fixed-Focus*. [B.r.]. Dostupné také z: [https://docs.luxonis.com/projects/hardware/en/latest/pages/guides/af\\_ff/#auto-focus-vs-fixed-focus](https://docs.luxonis.com/projects/hardware/en/latest/pages/guides/af_ff/#auto-focus-vs-fixed-focus).
25. LUXONIS. *OAK-D-Lite*. [B.r.]. Dostupné také z: <https://docs.luxonis.com/projects/hardware/en/latest/pages/DM9095/>.
26. LUXONIS. *DepthAI API*. [B.r.]. Dostupné také z: <https://github.com/luxonis/depthai>.
27. KUHNERT, Dennys. *Hand Physics Lab*. Holonautic, 2020. Dostupné také z: <https://sidequestvr.com/app/750/hand-physics-lab>.
28. SARKAR, Ayanava; PATEL, Ketul Arvindbhai; GANESH RAM, R.K.; CAPOOR, Geet Krishna. Gesture control of drone using a motion controller. In: *2016 International Conference on Industrial Informatics and Computer Systems (IIICS)*. 2016, s. 1–5. Dostupné z DOI: 10.1109/IIICSII.2016.7462401.



29. INTEL. *OpenCV: Open Source Computer Vision Library*. [B.r.]. Dostupné také z: <https://github.com/opencv/opencv>.
30. GOOGLE. *MediaPipe*. [B.r.]. Dostupné také z: <https://github.com/google/mediapipe>.
31. GOOGLE. *MediaPipe Hand landmarks detection guide*. 2023. Dostupné také z: [https://developers.google.com/mediapipe/solutions/vision/hand\\_landmarker](https://developers.google.com/mediapipe/solutions/vision/hand_landmarker).
32. MEDIAPIPE. *Hand Landmarks*. [B.r.]. Dostupné také z: <https://mediapipe.readthedocs.io/en/latest/solutions/hands.html>.
33. INDRIANI; HARRIS, Moh.; AGOES, Ali Suryaperdana. Applying Hand Gesture Recognition for User Guide Application Using MediaPipe. In: *Proceedings of the 2nd International Seminar of Science and Applied Technology (ISSAT 2021)*. Atlantis Press, 2021. ISBN 978-94-6239-451-3. ISSN 2352-5401. Dostupné z DOI: 10.2991/aer.k.211106.017.
34. KHROSOS GROUP. *OpenXR Plugin*. [B.r.]. Dostupné také z: <https://docs.unity3d.com/Packages/com.unity.xr.openxr@1.8/manual/index.html>.
35. ULTRALEAP. *Ultraleap Unity Plugin*. [B.r.]. Dostupné také z: <https://docs.ultraleap.com/xr-and-tabletop/xr/unity/index.html>.
36. LUXONIS. *Hand tracking with DepthAI – Host mode vs Edge mode*. [B.r.]. Dostupné také z: [https://github.com/geaxgx/depthai\\_hand\\_tracker#host-mode-vs-edge-mode](https://github.com/geaxgx/depthai_hand_tracker#host-mode-vs-edge-mode).
37. LUXONIS. *Hand tracking with DepthAI – Solo mode vs Duo mode*. [B.r.]. Dostupné také z: [https://github.com/geaxgx/depthai\\_hand\\_tracker#solo-mode-vs-duo-mode](https://github.com/geaxgx/depthai_hand_tracker#solo-mode-vs-duo-mode).
38. LUXONIS. *DepthAI – OAK For Unity*. [B.r.]. Dostupné také z: <https://github.com/luxonis/depthai-unity>.
39. GERARD-LUXONIS. *OAK For Unity – Spatial AI meets the power of Unity*. 2023. Dostupné také z: <https://forum.unity.com/threads/oak-for-unity-spatial-ai-meets-the-power-of-unity.1205764/#post-8946294>.
40. UNITY TECHNOLOGIES. *Python for Unity Manual*. [B.r.]. Dostupné také z: <https://docs.unity3d.com/Packages/com.unity.scripting.python@2.0/manual/index.html>.
41. ULTRALEAP. *LeapMotion Coordinate System*. [B.r.]. Dostupné také z: <https://developer.leapmotion.com/documentation/v4/concepts.html>.
42. ULTRALEAP. *Ultraleap Unity Api – class Leap.Frame*. [B.r.]. Dostupné také z: [https://docs.ultraleap.com/api-reference/unity-api/class/class\\_leap\\_1\\_1\\_frame.html#class-Leap.Frame](https://docs.ultraleap.com/api-reference/unity-api/class/class_leap_1_1_frame.html#class-Leap.Frame).
43. ULTRALEAP. *Ultraleap Unity Api – class Leap.Controller*. [B.r.]. Dostupné také z: [https://docs.ultraleap.com/api-reference/unity-api/class/class\\_leap\\_1\\_1\\_controller.html#unityclass\\_leap\\_1\\_1\\_controller](https://docs.ultraleap.com/api-reference/unity-api/class/class_leap_1_1_controller.html#unityclass_leap_1_1_controller).
44. ULTRALEAP. *Ultraleap Unity Api – class Leap.Unity.Provider*. [B.r.]. Dostupné také z: [https://docs.ultraleap.com/api-reference/unity-api/class/class\\_leap\\_1\\_1\\_unity\\_1\\_1\\_leap\\_provider.html?highlight=leaprovider](https://docs.ultraleap.com/api-reference/unity-api/class/class_leap_1_1_unity_1_1_leap_provider.html?highlight=leaprovider).
45. ULTRALEAP. *Stereo IR 170 Camera Module Evaluation Kit Datasheet*. [B.r.]. Dostupné také z: [https://www.ultraleap.com/datasheets/Stereo\\_IR\\_170\\_datasheet.pdf](https://www.ultraleap.com/datasheets/Stereo_IR_170_datasheet.pdf).
46. ULTRALEAP. *Ultraleap Unity Api – class Leap.Hand*. [B.r.]. Dostupné také z: [https://docs.ultraleap.com/api-reference/unity-api/class/class\\_leap\\_1\\_1\\_hand.html#unityclass\\_leap\\_1\\_1\\_hand](https://docs.ultraleap.com/api-reference/unity-api/class/class_leap_1_1_hand.html#unityclass_leap_1_1_hand).
47. UNITY TECHNOLOGIES. *Unity Documentation – Transform*. [B.r.]. Dostupné také z: <https://docs.unity3d.com/ScriptReference/Transform.html>.
48. UNITY TECHNOLOGIES. *Unity Documentation – Time.deltaTime*. [B.r.]. Dostupné také z: <https://docs.unity3d.com/ScriptReference/Time-deltaTime.html>.

49. GAME ANIME. *Simple Drone Unity Asset*. 2021. Dostupné také z: <https://docs.unity3d.com/ScriptReference/Time-deltaTime.html>.
50. SYNTY STUDIOS. *Simple Town – Cartoon Assets Demo Scene*. [B.r.]. Dostupné také z: [https://syntystore.com/products/simple-town-cartoon-assets?\\_pos=1&\\_psq=simple+town&\\_ss=e&\\_v=1.0](https://syntystore.com/products/simple-town-cartoon-assets?_pos=1&_psq=simple+town&_ss=e&_v=1.0).
51. UNITY TECHNOLOGIES. *Unity Cinemachine Package*. [B.r.]. Dostupné také z: <https://docs.unity3d.com/Packages/com.unity.cinemachine@2.9/manual/index.html>.
52. UNITY TECHNOLOGIES. *Unity Documentation – Collider*. [B.r.]. Dostupné také z: <https://docs.unity3d.com/ScriptReference/Collider.html>.
53. UNITY TECHNOLOGIES. *Unity Documentation – Rigidbody*. [B.r.]. Dostupné také z: <https://docs.unity3d.com/ScriptReference/Rigidbody.html>.
54. SOJKA, Matyáš. *Ovládání virtuálního dronu pomocí hand trackingu (tutorial)*. 2024. Dostupné také z: <https://www.youtube.com/watch?v=wnsRnt5YiAE>.
55. BAILEY, Hugh. *OBS Studio – Open Broadcaster Software*. [B.r.]. Dostupné také z: <https://obsproject.com/>.
56. SYNTY STUDIOS. *POLYGON – Prototype Pack Demo Scene*. [B.r.]. Dostupné také z: [https://syntystore.com/products/polygon-prototype-pack?\\_pos=1&\\_psq=protot&\\_ss=e&\\_v=1.0](https://syntystore.com/products/polygon-prototype-pack?_pos=1&_psq=protot&_ss=e&_v=1.0).
57. NOVÁČEK, Tomáš. *Technologie virtuální reality – Sledování polohy uživatele – možnosti*. CTU, Faculty of Information Technology, 2022. Dostupné také z: [https://docs.google.com/presentation/d/18f61UFvaLVsa6fuQLsCK4EFdFkyVEqWvrIGT2TJcIGc/edit#slide=id.gf1d867aaae\\_0\\_80](https://docs.google.com/presentation/d/18f61UFvaLVsa6fuQLsCK4EFdFkyVEqWvrIGT2TJcIGc/edit#slide=id.gf1d867aaae_0_80).
58. ULTRALEAP. *Ultraleap Tracking Software Download*. [B.r.]. Dostupné také z: <https://developer.leapmotion.com/tracking-software-download>.
59. SOJKA, Matyáš. *Ovládání dronu pomocí hand-trackingu*. 2024. Dostupné také z: <https://gitlab.fit.cvut.cz/sojkamat/ovladani-dronu-pomoci-hand-trackingu>.

# Obsah přiloženého média

readme.txt	.....	stručný popis obsahu média
exe	.....	adresář se spustitelnou formou implementace
├─ LEVEL_01	.....	první scéna s checkpointy
├─ LEVEL_02	.....	druhá scéna s checkpointy
└─ TUTORIAL	.....	výuková scéna
src	.....	zdrojové soubory
├─ impl	.....	odkaz na Unity projekt včetně kódů implementace
└─ thesis	.....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
text	.....	text práce
└─ thesis.pdf	.....	text práce ve formátu PDF