



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky

**Využití Kinect senzorů pro Motion Capture s programy
Source Filmmaker a Blender**

**Use of Kinect sensors for Motion Capture with Source
Filmmaker and Blender software**

Bakalářská práce

Vypracoval: Jan Vaněk

Vedoucí práce: PaedDr. Petr Pexa, Ph.D

České Budějovice 2022

Zadání bakalářské práce

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta

Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Jan VANĚK
Osobní číslo: P19454
Studijní program: B7507 Specializace v pedagogice
Studijní obor: Informační technologie a e-learning
Téma práce: Využití Kinect senzorů pro Motion Capture s programy Source Filmmaker a Blender
Zadávající katedra: Katedra informatiky

Zásady pro vypracování

Cílem bakalářské práce bude zpracovat možnosti využití Kinect senzorů pro Motion Capture v programech Source Filmmaker a Blender. Motion Capture je technologie pro zachycení pohybu pomocí snímání senzorů Kinectu a aplikování získaných dat v programech pro práci s 3D grafikou, bude tedy primárně zjišťován a následně popsán přesný postup, jak zpracovat získané údaje z Kinect senzorů v programech Blender a Source Filmmaker. V práci bude dále provedeno porovnání vhodnosti užití této techniky oproti klasické ruční animaci, popsány výhody a nevýhody využití v jednotlivých programech a budou otestovány možnosti kombinace Motion Capture a klasické animace. K tomu bude také nutné provést modifikace hardwaru Kinectů pro Xbox One, aby bylo možné připojit zařízení k počítači s Windows. Dále bude provedeno porovnání kvality snímání Kinectů určených pro Windows a konzolových Xbox One Kinectů a porovnání kvality snímání jedním a více Kinect senzory. Praktickou část práce budou tvořit vlastní animované sekvence, realizované pomocí Kinectů v kombinaci s ruční animací v Blenderu a tyto sekvence budou použity v Unity pro vytvoření interaktivního snímku, který bude moci uživatel sám ovládat.

Rozsah pracovní zprávy: 40
Rozsah grafických prací: CD ROM
Forma zpracování bakalářské práce: tištěná

Seznam doporučené literatury:

1. VAN GUMSTER, Jason. Blender for dummies. 4th edition. Hoboken: John Wiley, [2020]. For dummies. ISBN 1119616964.
2. HOLAN, Tomáš. Unity: první seznámení s tvorbou počítačových her. Praha: CZ.NIC, z.s.p.o., 2020. CZ.NIC. ISBN 978-80-88168-57-7.
3. Unity Real-Time Development Platform | 3D, 2D VR & AR Engine [online]. Unity Real-Time Development Platform | 3D, 2D VR & AR Engine. Copyright © 2021 Unity Technologies [cit. 05.04.2021]. Dostupné z: <https://unity.com/>
4. MSDN forums – Kinect for Windows v2 SDK [online]. Copyright © 2021 Microsoft [cit. 05.04.2021]. Dostupné z: <https://social.msdn.microsoft.com/Forums/en-US/home?forum=kinectv2sdk>
5. Kinect SDK tutorials & articles – developer Fusion [online]. Developer Fusion – ASP.NET, C# Programming, VB.NET, .NET Framework, Java and Visual Basic Tutorials. Copyright © [cit. 05.04.2021]. Dostupné z: <https://www.developerfusion.com/t/kinect/tutorials/>

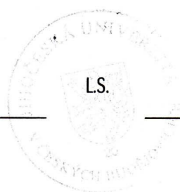
Vedoucí bakalářské práce: PaedDr. Petr Pexa, Ph.D.
Katedra informatiky

Zadání bakalářské práce

Datum zadání bakalářské práce: 7. dubna 2021
Termín odevzdání bakalářské práce: 30. dubna 2022



doc. RNDr. Helena Koldová, Ph.D.
děkanka



doc. PaedDr. Jiří Vaniček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 7. dubna 2021

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury. Prohlašuji, že v souladu s 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 20. dubna 2022

Jan Vaněk

.....

Abstrakt / Anotace

Cílem bakalářské práce bude zpracovat možnosti využití Kinect senzorů pro Mo-cap v programech Source Filmmaker a Blender. Motion Capture je technologie pro zachycení pohybu pomocí snímání senzorů Kinectu a aplikování získaných dat v programech pro práci s 3D grafikou, bude tedy primárně zjišťován a následně popsán přesný postup, jak zpracovat získané údaje z Kinect senzorů v programech Blender a Source Filmmaker. V práci bude dále provedeno porovnání vhodnosti užití této techniky oproti klasické ruční animaci, popsány výhody a nevýhody využití v jednotlivých programech a budou otestovány možnosti kombinace Mo-cap a klasické animace. K tomu bude také nutné provést modifikaci hardwaru Kinectů pro Xbox One, aby bylo možné připojit zařízení k počítači s Windows. Dále bude provedeno porovnání kvality snímání Kinectů určených pro Windows a konzolových Xbox One Kinectů a porovnání kvality snímání jedním a více Kinect senzory. Praktickou část práce budou tvořit vlastní animované sekvence, realizované pomocí Kinectů v kombinaci s ruční animací v Blenderu a tyto sekvence budou použity v Unity pro vytvoření interaktivního snímku, který bude moci uživatel sám ovládat.

Klíčová slova

Kinect, Motion Capture, SFM, Source Filmmaker, Blender, Unity, animace, 3D modely, grafika, iPiSoft, Mo-cap, fbx, dmx, bhv

Abstract/Anotation

The aim of my bachelor thesis is process the possibilities of using Kinect sensors for Motion Capture in Source Filmmaker and Blender programs. Motion Capture is a technology that captures motion by using Kinect sensors and applying the obtained data in 3D graphics programs. My main goal will be primarily to explore on subsequently describe the exact procedure to process the data obtained from Kinect sensors in Blender and Source Filmmaker. Furthermore, my other goals are comparison of the use of this technique to classical manual animation, the advantages and disadvantages of use in individual programs will be described and the possibilities of combining Motion Capture and classical animation will be tested. I'll also try to modify Kinect hardware for Xbox One for direct connection to a Windows computer. Furthermore, a comparison of the quality of Motion Capture from Kinects for Windows and the Xbox One and comparison of capturing motion with one or more Kinect sensors. For the practical part I will create my own animated sequences, captured by Kinects in combination with manual animation in Blender and these sequences will be used in Unity to create an interactive movie that the user himself will be able to control.

Keywords

Kinect, Motion Capture, SFM, Source Filmmaker, Blender, Unity, animation, 3D models, graphics, iPiSoft, Mo-cap, fbx, dmx, bhv

Poděkování

Velice děkuji vedoucímu své bakalářské práce panu PaedDr. Petru Pexovi, PhD. za jeho pomoc a cenné rady. Dále děkuji své úžasné rodině za jejich trpělivost a podporu a speciální poděkování svým prarodičům, bez kterých by se můj život dávno ubíral zcela odlišným směrem.

Obsah

1 Úvod	12
1.1 Východiska	12
1.2 Cíle	13
1.3 Metody	14
1.3.1 Teoretická část	14
1.3.2 Praktická Část	14
2 Teoretická část	16
2.1 Seznámení s Motion Capture	16
2.1.1 Historie	16
2.2 Základní typy Mo-cap systémů	17
2.2.1 Optické Mo-cap systémy	17
2.2.2 Neoptické Mo-cap systémy	17
2.2.3 Hybridní Mo-cap systémy	18
2.3 Důvod využití Kinectů	18
2.4 Druhy Kinectů	19
2.4.1 Kinect v1	21
2.4.2 Kinect v2	23
2.4.3 Kinect Azure	24
2.4.4 Kinect for Windows edice	25
2.5 Zvolený software	26
2.5.1 iPi Recorder 4 a iPi Mocap Studio 4	27
2.5.2 Source Filmmaker (SFM)	29
2.5.3 Blender	29
2.5.4 Unity	30
3 Příprava Mo-cap studia	31
3.1 Výběr hardwaru	31
3.1.1 Cena vytvoření domácího Mo-cap studia	33
3.2 Výběr snímacího prostoru	35
3.3 Příprava místnosti	35
3.4 Rozmístění senzorů	36

3.4.1	Rozmístění dvou senzorů	36
3.4.2	Rozmístění tří senzorů	37
3.4.3	Rozmístění čtyř senzorů	37
3.4.4	Vliv světla	37
3.4.5	Vliv oblečení herce a pozadí	38
3.4.6	Vliv vzdálenosti	38
3.4.7	Označení snímané plochy	39
3.5	Zapojení senzorů a motion controllerů	40
3.5.1	Zapojení Kinect v1 senzorů	40
3.5.2	Zapojení jednoho Kinect v2 senzoru	42
3.5.3	Zapojení více Kinect v2 senzorů do více počítačů	43
3.5.4	Nastavení master-slave v iPi Recorder 4	44
3.5.5	Manuální synchronizace nahraných sekvencí	47
3.5.6	Zapojení více Kinect v2 senzorů do jednoho počítače	48
3.5.7	Zapojení do jednoho počítače vs. Master-Slaves mode	49
3.5.8	Spárování motion controllerů	49
4	HW úprava Kinectu v2	52
4.1	Důvod úpravy Kinect senzoru	52
4.2	Jak úprava funguje	53
4.3	Rizika navrhované úpravy	55
4.4	Potřebné komponenty a nástroje	56
4.5	Proces úpravy Kinectu	57
4.5.1	Možné problémy připojení Kinectu po úpravě k počítači	68
4.5.2	Proč nepájet napájecí zdroj přímo na desku	69
4.6	Zhodnocení vhodnosti úpravy Kinectu	69
5	Snímání sekvencí	71
5.1	Získání dat pro porovnání s vypovídající hodnotou	71
5.2	Zvolené konfigurace pro porovnávání	72
5.3	Kombinace využití Kinectů v1 a v2	73
5.4	Vybrané typy pohybu pro porovnávání	73
5.5	Příprava herce	74

5.6	Nastavení nahrávání	77
5.7	Kalibrace motion controllerů	78
5.8	Evaluace pozadí scény	79
5.9	Nastavení nahrávání	80
5.10	Kalibrace scény	81
	5.10.1 Kalibrace pomocí 3D plochy	81
	5.10.2 Kalibrace pomocí světelné značky	86
	5.10.3 Zhodnocení kalibrace	87
5.11	T-Pose	88
5.12	Snímání sekvencí pro účely porovnávání	89
	5.12.1 Snímání s jedním Kinect senzorem	89
	5.12.2 Snímání se dvěma Kinect senzory	91
5.13	Zpracování sekvencí v iPi Mocap Studiu	92
	5.13.1 Sekvence s jedním Kinectem	92
	5.13.2 Sekvence s více Kinecty	98
5.14	Export animace do Blenderu	99
	5.14.1 Výběh 3D modelu	99
	5.14.2 Namapování na nový rig a export animace	101
	5.14.3 Vytvoření scény v Blenderu a export videa	102
6	Kvalita snímání	103
6.1	Minimalistický pohyb	103
	6.1.1 Jeden Kinect v1	103
	6.1.2 Dva Kinecty v1	103
	6.1.3 Jeden Kinect v2	104
	6.1.4 Dva Kinecty v2	104
	6.1.5 Celkové zhodnocení minimalistického pohybu	105
6.2	Normální pohyb	105
	6.2.1 Jeden Kinect v1	105
	6.2.2 Dva Kinecty v1	106
	6.2.3 Jeden Kinect v2	106
	6.2.4 Dva Kinecty v2	107
	6.2.5 Celkové zhodnocení normálního pohybu	107

6.3	Rychlý pohyb	108
6.3.1	Jeden Kinect v1	108
6.3.2	Dva Kinecty v1	108
6.3.3	Jeden Kinect v2	109
6.3.4	Dva Kinecty v2	109
6.3.5	Celkové zhodnocení rychlého pohybu	109
6.4	Otáčení	110
6.4.1	Jeden Kinect v1	110
6.4.2	Dva Kinecty v1	110
6.4.3	Jeden Kinect v2	111
6.4.4	Dva Kinecty v2	111
6.4.5	Celkové zhodnocení otáčení	112
6.5	Vzhůru nohama	112
6.5.1	Jeden Kinect v1	112
6.5.2	Dva Kinecty v1	113
6.5.3	Jeden Kinect v2	113
6.5.4	Dva Kinecty v2	114
6.5.5	Celkové zhodnocení snímání vzhůru nohama	114
6.6	Porovnání jednoho a dvou Kinectů	115
6.6.1	Snímání s jedním Kinectem	115
6.6.2	Snímání s více Kinecty	115
6.6.3	Zhodnocení	115
6.7	Porovnání snímání Kinectů v1 a v2	116
6.8	Vliv velikosti herce	116
6.9	Možnost snímání ve tmě	117
6.10	Zlepšení trackingu chodidel pomocí bot	118
6.11	Využití motion controllerů	118
6.12	Interakce s objekty	119
6.12.1	Sezení na židli	119
6.12.2	Interakce s malým předmětem	120
6.13	Snímání více herců	121

7.1	Export modelu do SFM	123
7.2	Export z iPi studia do SFM	132
7.3	Zpracování sekvencí v SFM	134
7.4	Možnosti úprav sekvence	135
7.5	Retargeting animace	137
7.6	Export z iPi studia do Blenderu	139
7.7	Zpracování sekvencí v Blenderu	140
7.8	Možnosti úprav importované sekvence	140
7.9	Retargeting animace	141
7.10	Blender vs. SFM	142
7.10.1	Zpracování surových sekvencí	142
7.10.2	Výhody a nevýhody zpracování v Blenderu a SFM	143
7.10.3	Motion Capture vs. ruční animace	144
8	Tvorba aplikace v Unity 3D	147
8.1	Import sekvencí do Unity Engine	147
8.2	Vývoj aplikace v Unity	149
8.3	Představení vytvořené aplikace	153
9	Tvorba tutoriálů a webu	155
9.1	Tvorba tutorial videí	155
9.2	Vytvoření webu howtomocap.com	157
10	Zhodnocení a závěr	161
	Seznam použité literatury a zdrojů	163
	Seznam obrázků	171
	Seznam tabulek	172
A	Příloha - DVD	173
B	Příloha - Web	174
C	Příloha - Youtube kanál	175

1 Úvod

Tato bakalářská práce se zabývá využitím Kinect senzorů pro Motion Capture (zkráceně Mo-cap) a zhodnocení kvality jejich snímání. V teoretické části je vysvětleno, co je vlastně Motion Capture, proč je vhodné zvolit pro začátečníky s Mo-cap právě Kinecty, jaké jsou rozdíly mezi jejich jednotlivými typy a jaký je rozdíl mezi Kinecty a běžnými kamerami. Také blíže představím software, který je v rámci této práce využíván.

V praktické části se budu věnovat hardwarové úpravě Kinect senzoru v2 určeného pro Xbox One, dále přípravě snímacího prostoru, zapojení hardwaru a jeho zprovoznění na počítačích s Windows 10, procesem kalibrace, snímání a zpracování získaných surových sekvencí, exportem zpracovaných sekvencí do 3D programů Source Filmmaker a Blender a možnostech jejich následné úpravy, využití v kombinaci s ruční animací, import animací do Unity a jejich využití při vývoji indie her.

V závěru vyhodnotím kvalitu získaných sekvencí a představím svůj web, kde budou mé výsledky publikovány pro veřejnost spolu se sérií mnou vytvořených tutoriálů zaměřených na využití Kinect senzorů pro Mo-cap.

1.1 Východiska

Se stále se zvyšujícím výkonem domácích počítačů se běžným uživatelům otevírají zcela nové možnosti, které dříve patřily pouze do hal profesionálních animačních studií. Za posledních 15 let se možnosti práce s 3D grafikou na dostupných počítačích posunuly natolik, že je možné na nich vytvářet poměrně působivé animované snímky a hry. Každý rok se objevují stovky až tisíce nových nadšenců do animace a vývoje indie her (hry většinou vytvářené menšími polo-profesionálními týmy nadšenců a vývojářů) po celém světě a využití Mo-cap v kombinaci s klasickou ruční animací může úroveň těchto projektů posunout o další úroveň výš.

Většina začínajících animátorů si pod pojmem Motion Capture pravděpodobně představí velké studio s obrovským rozpočtem, obří halu a herce v oblecích s hromadou senzorů a značek na obličejích. Je pravda, že cena vybudování profesionálního Mo-cap studia se pohybuje v řádech statisíců až milionů korun podle zvoleného

systémového řešení a prostoru pro vytvoření Mo-cap studia. Ovšem existují i možnosti, jak začít s Mo-cap s mnohem menším rozpočtem v řádech tisíců až deseti-tisíců korun a dosáhnout poměrně dobrých a použitelných výsledků.

Využití Kinectů se pro tento účel přímo nabízí a to hned z několika důvodů. Kinecty nejsou normální barevné kamery, ale mají v sobě také hloubkový senzor, který při Mo-cap odvádí relativně dobrou práci, protože snímá hloubková data, pomocí kterých určuje, kde v prostoru se nacházíte. Jedná se o tzv. RGB-D senzor, který snímá barevná i hloubková data. [1] Druhý důvod je, že Kinect určený pro konzoli Xbox 360 či novější verzi určenou pro Xbox One má spousta lidí již doma a stačí jim zakoupit adaptér pro PC, aby mohli s domácím Mo-cap začít.

Téma využití Kinectů pro Mo-cap je sice známé, ale možnosti jeho využití napříč různými 3D programy již tak dobře zdokumentované není. Většina informací, které se dají ohledně tohoto tématu nalézt, jsou jen kusé, mohou být nepřesné a zavádějící a pro začínajícího animátora, který se chce do Mo-cap s Kinecty vrhnout, to může být odrazující. Proto věřím, že vytvoření webu ohledně této zajímavé problematiky, kde zkonduzuji veškeré své získané poznatky na jednom místě, mohou být všem začínajícím animátorům přínosem.

Vzhledem k tomu, že majoritní část amatérských animátorů začíná s programy jako je Source Filmmaker nebo Blender, a také kvůli tomu, že tyto programy jsou zdarma, jsem se rozhodl prozkoumat možnosti využití nasnímaných sekvencí právě v těchto dvou programech a možnosti jejich zpracování v nich. Porovnání vhodnosti užití mezi těmito dvěma programy pro Mo-cap data z Kinectů jsem nikde jinde nenalezl, a proto ho v rámci této práce provedu také.

1.2 Cíle

Tato bakalářská práce má několik dílčích cílů. Hlavním cílem je přiblížit možnosti využití Kinectů pro Mo-cap širšímu publiku. Dalším z cílů je důkladné prozkoumání kvality snímání Kinectů, porovnání využití starších typů Kinect senzorů v1 oproti vylepšeným Kinect senzorům v2. Chci zjistit možnosti snímání více Kinecty najednou a výhod, které taková konfigurace přináší. Prozkoumat možnost úpravy konzolových Kinectů v2 určených pro Xbox One, abych se obešel i bez stále hůř dostupných adaptérů. Podívat se na rozsah využití získaných sekvencí a jejich

zpracování v programech Source Filmmaker (SFM) a Blenderu a použití získaných sekvencí při vývoji aplikace v Unity.

Cíle přiblížení problematiky využití Kinectů pro Mo-cap hodlám dosáhnout vytvořením webu, který se bude tomuto tématu věnovat, a kde budu publikovat své získané poznatky, a také vytvořením série video tutoriálů, kde podrobně popíšu postup práce s Kinecty při získávání sekvencí a jejich zpracování. Průzkum kvality snímání v praktické části hodlám provést pomocí snímání různých typů pohybu s rozdílnými typy senzorů a jejich různým počtem a tyto získané sekvence následně proti sobě vyhodnotit.

Součástí práce bude vytvoření aplikace v Unity za pomoci nasnímaných sekvencí, čímž hodlám ověřit vhodnost využití této metody při vývoji indie her.

1.3 Metody

1.3.1 Teoretická část

V teoretické části seznámím čtenáře s tím, co je vlastně Motion Capture. Vysvětlím čtenáři, proč jsem si pro něj zvolil právě Kinecty, zaměřím se na rozdíly mezi jednotlivými typy Kinect senzorů a jaký je rozdíl mezi použitím Kinectů, klasických kamer a celotělových obleků. Vysvětlím některé limitace Kinect senzorů a možnosti, jak je kompenzovat. Podívám se na software, který budu v práci využívat a hlavní důvody, proč jsem si ho vybral. Provedu porovnání nákladů na sestavení základního Mo-cap studia s Kinect senzorem a profesionálním řešením.

1.3.2 Praktická Část

V praktické části předvedu hardwarovou úpravu konzolové verze Kinectu v2 určené pro Xbox One a upravím jeho napájení a připojení k PC tak, aby šel připojit klasickým USB kabelem bez použití adaptéru. Poté se zaměřím na přípravu snímacího prostoru a optimální podmínky pro snímání. Prozkoumám možnosti připojení všech verzí Kinectů k PC pro použití se softwarem iPiSoft, nastavení dílčího snímání několika počítači v režimu Master-Slaves, podmínky připojení více senzorů k jednomu PC. Zjistím, jak velkou roli hraje výkon počítače při zpracování sekvencí v iPiSoft Mo-cap studiu a při nahrávání sekvencí.

Nasnímám a zpracuji různé typy pohybu s různými druhy Kinectů a jejich různým počtem a následně vyhodnotím jejich kvalitu, zjistím jakou roli hraje počet snímajících senzorů a rozdíl v kvalitě snímání Kinect senzory v1 a Kinect senzory v2. Následně předvedu různé možnosti exportu sekvencí do programů Source Filmmaker a Blender a na různé typy bipedálních rigů. Podívám se na možnosti práce se sekvencemi a jejich úpravy a kombinaci s ruční animací v SFM a Blenderu a proberu výhody a nevýhody jejich zpracování v těchto programech. Dále sekvence importuji do Unity 3D a vytvořím s jejich pomocí menší aplikaci.

Závěrem představím svůj web, kde budou publikované mé poznatky a série video tutoriálů zaměřujících se na Mo-cap s pomocí Kinectů.

2 Teoretická část

2.1 Seznámení s Motion Capture

Zjednodušeně řečeno, Motion Capture je technika, při které se zachytí pohyb a mimika herce, a přeneše se jako informace pro užití ve filmech a videohrách. V dnešní době se prakticky žádná velká videohra ani film využívající CGI bez Mo-cap neobejde. Jako příklad můžu uvést ikonickou postavu Gluma z trilogie Pána Prstenů, jehož ztvárnil herec Andy Perkis. Jeho použitím se dá dosáhnout mnohem reálnějšího a přirozenějšího pohybu, než klasickými animačními metodami.

2.1.1 Historie

Počátky moderního Motion Capture sahají nejspíš až do let 1914-1916, kdy animátor Max Fleisher se svým bratrem začali provádět testy s technikou rotoskopie, kterou si nechal roku 1917 Max Fleisher patentovat. Tato technika spočívá v překreslování natočeného materiálu animátorem snímek po snímku. [2] Původně se natočený materiál promítal na skleněnou desku, na které následně animátor snímky překresloval. Tato metoda byla velice pomalá, protože animátor musel ručně překreslovat každý jednotlivý snímek.

V roce 1959 si animátor Lee Harrison udělal oblek se zabudovanými potenciometry a s jejich pomocí dokázal snímat pohyb herce v obleku v reálném čase na CRT obrazovce. Jednalo se pravděpodobně o první výskyt Mo-cap v reálném čase.[2]

Moderní Motion Capture je v podstatě velmi pokročilá forma rotoskopie. Mo-cap jak se používá dnes, se začal objevovat v osmdesátých letech dvacátého století. Massachusetts Institute of Technology vyvinul svojí takzvanou "grafickou loutku". Tento systém fungoval tak, že se na oblek herce přidělala LED světla a snímá se jeho pohyb pomocí systému Op-eye. Tento systém využíval dvě a více kamer. Jednalo se pravděpodobně o první funkční optický Mo-cap systém. Ze záběrů z kamer specializovaný software z pozic led světel dokázal propočítat jejich umístění ve 3D prostoru v každém snímku.[3] [4]

Následný vývoj Mo-cap byl po nějakou dobu utlumený kvůli hardwarovým nárokům a limitacím výkonu tehdejších počítačů. I v devadesátých letech kamery

používané pro Mo-cap dosahovaly velikosti lednic a animátoři stále museli v každém jednotlivém snímku manuálně přiřazovat každou značku.[2] A počítače, které by byly schopné automaticky propočítat komplexní pohyb herců mezi snímky, neexistovali. Po roce 2000 nastal pro Mo-cap boom a jeho technologie se začala rapidně vyvíjet a tento vývoj pokračuje dodnes.

2.2 Základní typy Mo-cap systémů

Mo-cap systémy se dají rozdělit do třech základních kategorií. Jsou to optické Mo-cap systémy, neoptické Mo-cap systémy a hybridní Mo-cap systémy.

2.2.1 Optické Mo-cap systémy

Optické systémy jsou založené na principu snímání z několika kamer, ze kterých software vyhodnocuje pro každý časový úsek aktuální pozici herce.

Optické systémy se dále dělí na dvě podkategorie. První jsou značkové optické systémy (Marker Motion Capture systems), které vyžadují, aby herec měl na těle různé druhy označení, které kamery zachycují, a vyhodnocují jejich pozici a rotaci. Značkové Mo-cap systémy se dají rozdělit do čtyř podkategorií. [5]

1. Aktivní značky (Active markers)
2. Časově modulované aktivní značky (Time modulated active markers)
3. Pasivní značky (Passive markers)
4. Polo-pasivní nepostřehnutelné značky (Semi-passive imperceptible markers)

Druhé jsou neznačkové Mo-cap systémy. Ty nevyžadují žádné značky umístěné na těle herce ani žádné speciální obleky. Herec může nosit klasické oblečení. Tento systém je založen na technologii počítačového vidění a rozpoznávání vzorců. Silueta herce je zkoumána několika kamerami z různých úhlů. [5]

2.2.2 Neoptické Mo-cap systémy

Neoptické Mo-cap systémy se dělí do tří základních kategorií podle typu senzoru, který využívají.

1. Inerciální (Inertial)
2. Mechanické (Mechanical)
3. Magnetické (Magnetic)

Neoptické systémy nevyužívají kamery, ale data se získávají přímo ze senzorů, které po kalibraci systému udávají údaje o poloze a rotaci dané části herce, na kterou byl senzor umístěn.[5]

2.2.3 Hybridní Mo-cap systémy

Hybridní systémy využívají kombinaci předchozích systémů, aby kompenzovaly nedostatky, které jednotlivé systémy mají. Jedná se o pravděpodobně nejčastěji aktuálně užívaný typ Mo-cap systému a budu ho užívat i v rámci této práce. Každý z předchozích systémů má nějakou slabinu. Například u optických systémů je důležité, aby byl herec neustále viděn kamerami, jinak není možné jeho pohyb sledovat. U některých senzorů po určitém časovém úseku selhává kalibrace a musí se provádět znova, jinak se projeví drift, magnetické mohou rušit kovové objekty a tak dále.[5][6]

Výhody a nevýhody těchto systémů podrobněji zkoumala Miroslava Jarešová v roce 2012 ve své práci „Zachycení a vizualizace pohybu v reálném čase“ [6] a proto se podrobnějšímu popisu těchto systémů v rámci této práce věnovat nebudu.

2.3 Důvod využití Kinectů

První verze Kinectu pro konzoli Xbox 360 se dostala na pulty obchodů v roce 2010, v roce 2012 Microsoft vydal verzi určenou pro Windows PC. O rok později se objevila značně vylepšená verze původního senzoru pro konzoli Xbox One a v roce 2014 Microsoft vydal tuto vylepšenou verzi i ve variantě pro PC.

Výroba Kinect senzorů byla ukončena v roce 2017 a za dobu od vydání se prodalo přes 35 milionů Kinect senzorů.[7] I když se nové senzory pro Xbox 360 a Xbox One ani jejich varianty pro PC již nevyrábí, stále jsou lehce dostupné a hlavně je mají miliony lidí stále doma a mohou s nimi zkusit Mo-cap hned.

Případně si mohou pořídit novou vylepšenou verzi Kinect Azure, kterou Microsoft oznámil v roce 2019 a vyšla v březnu 2020 pouze ve verzi pro PC.[8]

Kinect senzory jsou vybaveny jak klasickou barevnou kamerou, tak hloubkovým senzorem, který vytvoří hloubkovou mapu analýzou vzoru skvrnitosti. Tyto data potom specializovaný software dokáže zpracovat a přenést do virtuálního 3D prostoru a namapovat je na rig 3D modelu. Z klasických barevných kamer se získá pouze 2D obraz postrádající hloubku a algoritmus poté musí propočítávat a odhadnout hloubku posouzením výstupu jednotlivých kamer a pozice siluety herce nebo jednotlivých značek v jednotlivých snímcích záznamu. Z hloubkového senzoru Kinectu se získá skutečný 3D obraz herce, což se může projevit větší přesností.

Díky velkému úhlu, který Kinecty zachycují, je možné sestavit si domácí Mo-cap studio i v poměrně malé místnosti, jejich kalibrace není nijak složitá a je vyžadována jen jednou za nahrávací sezení, pokud se senzory není pohybováno. Proto jsou do začátku ideální volbou pro ty, kdo si chtějí postavit levné Mo-cap studio přímo u sebe doma.

Jejich nevýhodou je, že se nedají použít venku. Mají poměrně krátkou vzdálenost, na kterou se s nimi dá snímat (okolo 4 metrů, jak jsem si sám ověřoval při jejich používání). Takže na velké scény, kde je potřeba, aby se herci hodně pohybovali, jsou nevhodné. Jako absolutně největší nevýhodu osobně vnímám, že jsou limitované na 30 snímků za vteřinu [9], což u rychlých pohybů může vést k problémům se snímáním.

2.4 Druhy Kinectů

Existují tři základní druhy Kinect senzorů.

1. **Kinect v1** - Původní Kinecty určené pro Xbox360 a jejich verze pro PC, které se většinou označují jako v1.
2. **Kinect v2** - Druhý vylepšený typ určený pro Xbox One a jeho PC verze, který se většinou označuje jako v2.
3. **Kinect Azure** - Nejnovější vylepšená verze Kinect Azure, která je dostupná pouze pro PC a již nemá konzolovou variantu.

Mezi jednotlivými verzemi jsou zásadní rozdíly v rozlišení kamer, snímanému úhlu, snímané vzdálenosti i technologii snímání, jak ukazuje tabulka.

	Kinect v1	Kinect v2	Kinect Azure
Rozlišení RGB kamery	1280 x 720 px 12 fps 640 x 480 px 30 fps	1920 x 1080 px 30 fps	3840 x 2160 px 30 fps
Rozlišení hloubkové kamery	320 x 240 px 30 fps	512 x 524 px 30 fps	NFOV unb - 640 x 576 px 30 fps NFOV bin - 320x288 px 30 fps WFOV bin - 512x512 px 30 fps WFOV unb - 1024x1024 px 15 fps
Technologie hloubkového snímání	Structured light-pattern projection	ToF (Time-of-Flight)	ToF (Time-of-Flight)
Zorné pole hloubkového senzoru	57 ° x 43 ° alt. 58.5 ° x 46.6 °	70 ° x 60 ° alt. 70.6 ° x 60 °	NFOV - 75 ° x 65 ° WFOV - 120 ° x 120 °
Aktivní snímací vzdálenost	0,4 - 4 m	0,5 - 4,5 m	NFOV unb-0,5 - 4,86 m NFOV bin - 0,5 - 5,46 m WFOV bin- 0,25 - 2,88 m WFOV unb - 0,25 - 2,21 m

Tabulka 1: Rozdíly senzorů [1]



Obrázek 1: Kinect senzor v1 (nahore) a v2 (dole)

2.4.1 Kinect v1

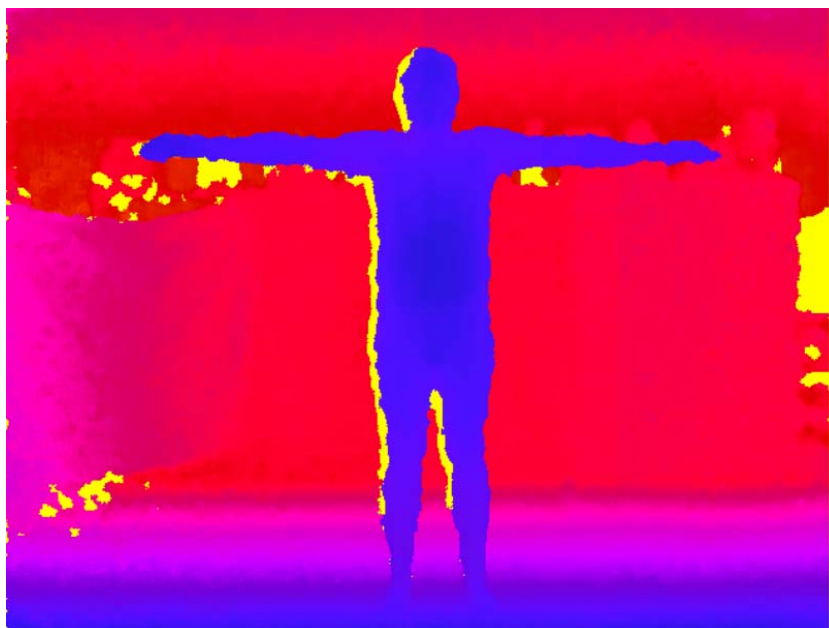
Původní verze Kinectu v1 je lehčí a o pár centimetrů širší než jeho nástupce v2, tato verze také postrádá možnost přidělení na stativ, což jeho užití pro Mo-cap může lehce komplikovat. Jeho RGB kamera s rozlišením 640 x 480 pixelů při 30ti fps a zorným polem 62 ° x 48,6 ° dává zhruba 10 x 10 pixelů na stupeň. Jeho hloubková kamera 320 x 240 pixelů a zorným polem 58,5 ° x 46,6 ° stupně dává průměrně 5 x 5 pixelů na stupeň. Používá starší technologii hloubkového snímání (Structured light-pattern projection), jejímž výsledkem je interpolovaný hloubkový obraz, který je založen na mnohem nižším počtu vzorků, než jaké naznačuje rozlišení hloubkového obrazu. [13]

Tato verze se v České republice prodává v rozmezí od 700 - 2000 Kč, verze v1, určená pro Windows, je většinou o pár stovek korun dražší než konzolová verze. Dá se najít téměř na všech větších internetových bazarech a v některých obchodech jsou stále k prodeji nové nepoužité kusy. Adaptér pro zapojení konzolové verze senzoru do PC se běžně prodává v rozmezí mezi 400 - 500 Kč.

Jak je z jeho specifikací vidět, jedná se už o poměrně zastaralý kus hardwaru a rozlišení jeho kamer je pro dnešní dobu již bídné. Nicméně je možné ho pro Mo-cap



Obrázek 2: Kinect v1 RGB



Obrázek 3: Kinect v1 Depth

použít a pokud je velmi omezený rozpočet, jedná se o solidní základ, jak si práci s Mo-cap vyzkoušet. Osobně bych doporučil si trochu připlatit za v2. Já jsem tyto

senzory koupil, jen abych je mohl v rámci této práce vyzkoušet a porovnat kvalitu jejich snímání a vhodnost užití.

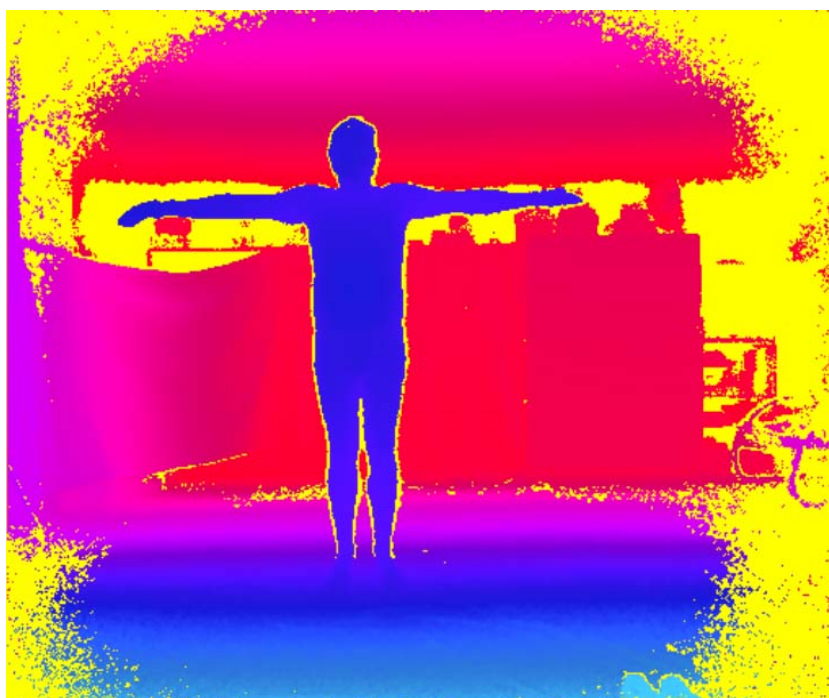
2.4.2 Kinect v2

Kinect v2 je oproti předešlé verzi těžší, skladnější a má zesponu možnost přidělení ke klasickému stativu. Jeho RGB kamera s rozlišením 1920 x 1080 pixelů při třiceti snímcích za vteřinu a zorným polem 84,1 ° x 53,8 ° dává zhruba 22 x 20 pixelů na stupeň. Což je výrazné zlepšení, pokud bych chtěl data z RGB kamery použít například pro klíčování se zeleným plátnem. Jeho hloubková kamera 512 x 424 pixelů a zorným polem 70,6 ° x 60 ° stupně dává průměrně 7 x 7 pixelů na stupeň. To se na první pohled může jevit jen jako mírné zlepšení, ovšem Kinect v2 používá novější technologii Time-of-Flight. Díky tomu každý pixel hloubkového obrazu obsahuje reálně naměřenou hodnotu hloubky (souřadnici z) a je mnohem preciznější než hloubková kamera předchozí verze Kinectu.[13]



Obrázek 4: Kinect v2 RGB

Tato verze je v České republice běžně k dostání v rozmezí od 2 500 - 3 500 Kč, dá se sehnat na většině známějších internetových bazarů a v různých internetových obchodech s konzolovým příslušenstvím. K dostání je téměř výhradně jen konzolová verze, vzhledem k velmi krátké produkci verze pro Windows. Ke konzolové verzi je potřeba využít adaptér, který se prodává v rozmezí mezi 800 - 2 000 Kč a začíná být nedostatkovým zbožím, protože už se nové nevyrobí. Také je zde



Obrázek 5: Kinect v2 Depth

možnost Kinect v2 upravit pro připojení k PC bez pomoci adaptéru, ale tato varianta vyžaduje zásah do hardwaru a tím kupující samozřejmě přijde o jakoukoliv záruku.

Tento model je výrazně lepší v každém ohledu oproti původnímu Kinectu v1 a osobně bych doporučil pořízení právě této verze, protože podle mě má nejlepší poměr cena/výkon. Získaná hloubková data jsou výrazně lepší a snímání pohybu je s ním přesnější.

2.4.3 Kinect Azure

Nejnovější verze Kinect senzoru pojmenovaná Azure je lehčí než v2. Jeho RGB kamera s rozlišením 3840 x 2160 pixelů při 30ti fps je nejlepší, jaká se na Kinectech dá najít. Na měření hloubky používá stejnou technologii jako v2 (Time-of-Flight) a hloubková kamera nabízí několik režimů nastavení od NFOV (Narrow field-of-view) binned s rozlišením 320 x 288 pixelů při 30ti fps po WFOV (Wide field-of-view) unbinned s rozlišením 1024 x 1024 px při 15ti fps. [1] I nejnovější Kinect je

limitovaný pouze na 30 snímků za vteřinu.

Kinect Azure se v běžném prodeji prakticky nevyskytuje a sehnat ho je velmi těžké. Na stránkách Microsoftu je k nalezení s cenovkou 399\$, ale je již dlouhodobě nedostupný k objednání. Dá se také nalézt v aukcích na Ebay a podobných portálech s cenovkami okolo 1 200\$.

Kinect Azure se mi bohužel nepodařilo sehnat, a proto se jím v této práci dále zabývat nebudu. S jeho aktuální cenou bych jeho koupi a použití pro levné domácí Mo-cap řešení stejně nedoporučil.

2.4.4 Kinect for Windows edice



Obrázek 6: Kinect for Windows edice

Microsoftem vydané edice Kinect senzorů pro Windows jsou hardwarově a funkčně téměř identické s jejich konzolovými variantami. Pro Mo-cap je možné použít obě a dosáhnout totožných výsledků. Kinect v1 určený pro Windows má navíc Near mode, který by měl lépe rozpoznávat objekty v bližší vzdálenosti okolo 50ti centimetrů. Rozdíly mezi konzolovým Kinectem v2 a tím, který je určený pro Windows je zřejmě jen v absenci loga Xbox na PC verzi. Informace o rozdílech

mezi konzolovými a PC verzemi nejsou snadno dostupné, ale z toho, co se mi podařilo najít, PC verze jsou v podstatě klasické Kinecty, které u sebe mají rovnou přibalený adaptér. [10][11]

Je podstatné si uvědomit, že není problém pro Mo-cap použít konzolovou verzi senzoru, protože verze určené pro Windows se shánějí mnohem hůř. To platí především pro Kinect v2, jehož PC verze byla vyráběna pouhý rok, než jí Microsoft zastavil a uživatelům doporučil použít konzolový Kinect s oficiálním adaptérem pro PC.[12]

2.5 Zvolený software

Při výběru softwaru pro Mo-cap studio byla rozhodující tato kritéria.

1. **Umožňuje snímání s Kinect senzory** - Vzhledem k tomu, že moje Mo-cap řešení je postavené na nich, byl tento bod naprosto nekompromisní.
2. **Kvalita zpracování a možnosti programu** - Různá softwarová řešení mohou nabízet různou kvalitu zpracování dat z Kinect senzorů a mohou nabízet další užitečné funkce. Například užívání motion controllerů jako senzorů pohybu dlaní a hlavy a vytvoření hybridního Mo-cap systému, možnosti exportu animovaných sekvencí do různých formátů, možnost aplikace na vlastní netradiční Rig a další.
3. **Cena** - Vzhledem k tomu, že se v rámci této práce snažím přiblížit Mo-cap co nejširšímu publiku, je důležité brát ohled i na cenu vybudování takového základního Mo-cap studia. Cena licence za Mo-cap software se může různit od několika stovek dolarů až po desítky tisíc dolarů. Je potřeba zvážit, kolik chce uživatel do softwaru investovat a co od něj očekává.
4. **Uživatelsky přívětivé ovládání** - Pro nové uživatele je důležité, aby bylo prostředí programu přehledné a snadno ovladatelné.
5. **Dostupnost dokumentace k užívání programu** - Software, ke kterému není kvalitní dokumentace a má malou základnu uživatelů, kteří na internetu nesdílejí své poznatky a rady, je mnohdy velmi těžké se naučit a pokud při

práci s programem narazím na nějaký problém, je malá šance, že snadno naleznu jeho řešení. Proto oblíbenost programu mezi uživateli podle mě hraje důležitou roli. Aktivní komunita uživatelů může být při práci takřka nedocenitelná, stejně tak jako kvalitně zpracovaná dokumentace od výrobce softwaru.

6. **Funkčnost softwaru na Windows 10** - Vzhledem k tomu, že tento systém je momentálně nainstalován na většině domácích počítačů, a já se chci zaměřit na domácí řešení Mo-cap studia, musí být software přizpůsobený na práci v systému Windows 10.

2.5.1 iPi Recorder 4 a iPi Mocap Studio 4

Jako software pro své Mo-cap řešení jsem zvolil dvojici programů iPi Recorder 4 a iPi Mocap Studio 4. Po zvážení ostatních alternativ (jako například Brekel nebo Ni Mate) mi ve většině bodů přišel jako nejlepší volba.

Podle typu zvolené licence dokáže najednou zpracovat data až ze čtyř hloubkových kamer v jedné scéně. Umožňuje snímání v režimu Master-Slaves pro připojení více Kinect senzorů v2, které jsou oficiálním Windows ovladačem limitované na připojení pouze jednoho senzoru k počítači (to lze obejít pomocí libfreenect2, což bude vysvětleno dále v rámci této práce). Umí pracovat s několika druhy motion controllerů a jejich výstupy použít pro zpracování pohybu a rotace rukou a hlavy. [15] To znamená, že je v něm možné vytvořit hybridní Mo-cap systém, kde zkombinuji optický systém založený na Kinectech s neoptickým systémem s použitím motion controllerů (Nintendo Joy-cons, PS move a dalších). Neumí sledovat pohyby prstů ani výraz tváře, ovšem nabízí možnost ručního klíčování pohybu prstů. [15]

Jedná se pravděpodobně o uživatelsky nejoblíbenější a velmi dobře zdokumentovaný software pro Mo-cap s Kinect senzory, na internetu je aktivní komunita uživatelů, která s ním dodnes pracuje a je k němu dostatek tutoriálů a kvalitně zpracovaná dokumentace od výrobce. K tomu iPiSoft software neustále vylepšuje a přidává podporu nových zařízení.

Jedná se vlastně o dva oddělené programy od iPiSoft, první je iPi Recorder 4, který je zcela zdarma a umožňuje nahrávání scény. Druhý je iPi Mocap Studio 4, který nabízí třicetidenní zkušební období zdarma, a poté je na výběr z několika

druhů licencí podle potřeb uživatele.

	Express Licence	Basic Licence	Pro Licence
Cena Licence	195\$ stálá	695\$ stálá 345\$ roční 165\$ tříměsíční	1 995\$ stálá 995\$ roční 495\$ tříměsíční
Počet možných hloubkových senzorů	1	1 - 2	1 - 4
Počet možných RGB kamer	0	3 - 6	3 - 16
Počet možných snímaných herců	1	1	1 - 3
Vhodné užití	jednoduché pohyby	přesný motion tracking	snímání více herců

Tabulka 2: Rozdíly v licencích iPiSoft Mocap Studio 4 [14]

Základní express licence vyjde na zhruba 4 500 Kč a na jednoduché snímání je zcela postačující. Pokud bych nechtěl být limitován jedním senzorem, stálá základní licence vyjde na zhruba 16 000 Kč a umožní provádět mnohem preciznější motion tracking [15] a tuto licenci jsem si zvolil pro své domácí Mo-cap studio.

Licence Pro jako jediná umožňuje snímání více herců najednou, ovšem pro řešení studia s Kinecty je podle mě zbytečná, protože vzhledem k omezené snímací vzdálenosti Kinectů vytvoří pouze malou snímací plochu, akorát pro jednoho herce.

Výhodou je, že můžu zachytit sekvence i s více senzory a herci pro potřeby porovnávání, protože iPi Recorder nemá v tomto ohledu licenční omezení, a dokonce je můžu i zpracovat v iPi Mocap studiu. Ovšem bez patřičné licence je není možné exportovat z programu.

2.5.2 Source Filmmaker (SFM)

Source Filmmaker (zkráceně SFM) je animační software od Valve využívající Source Game Engine. Jeho první verze byla vydaná pro veřejnost v roce 2012 a je zcela zdarma. Dodnes má na steamu každý den podle statistik 1500 - 2 000 uživatelů [16] a jedná se o jeden z nejoblíbenějších programů mezi začínajícími animátory.

Oproti ostatním programům pro práci s 3D grafikou má velmi omezené možnosti a zastaralý renderovací engine. Chybí například možnost upravit topologii modelů nebo možnost jednoduše přidat sekundární a terciární pohyb k animaci, protože nemá možnost rozložit animaci do vrstev.

Jeho výhody jsou, že je přehledný a extrémně jednoduchý na naučení a použití. Osobně jsem ho používal od roku 2014 a pro kohokoliv, kdo se chce naučit základy animace ve 3D, podle mě není žádná lepší možnost. I přes veškerá svá omezení se jedná o velmi solidní animační program, pokud jde skutečně pouze o animaci a tvoření animovaných snímků. Pokud uživatel potřebuje jednoduše exportovat animace do enginu jako je Unity nebo Unreal Engine, provádět simulace kapalin, fyziky a podobně, je SFM nedostatečné.

Pro SFM existuje obrovská knihovna modelů přímo od Valve z jejich her, které jsou všechny zdarma, a na Steamu ve svém workshopu má další tisíce modelů nahraných samotnými uživateli. Můžete do něj přidat velmi jednoduše i své vlastní modely a aplikovat upravené rigy.

Tento program jsem si zvolil především kvůli jeho jednoduchosti a oblíbenosti a doporučuji ho jako startovní bod, pokud uživatel nemá žádné předchozí zkušenosti s animacemi, riggingem a 3D modelováním.

2.5.3 Blender

Blender je mnohem pokročilejší než SFM ve všech směrech, nejedná se primárně o animační program, ale je v něm možné provádět vše od 3D modelování, riggingu, animace, texturování, uv mappingu a tak dále. Blender je zcela zdarma, i když některé jeho addony jsou placené, ty ovšem nejsou nutně potřeba. Blender je neustále vyvíjen a vylepšován a každý rok vychází několik jeho nových verzí a každá většinou přináší důležité změny a upgrady jednotlivých systémů. Jako například přidání real-time renderovacího enginu Eevee, vylepšení sculpt módu, De-noiser a

mnoho dalších funkcí.[17]

Co je jeho nevýhodou je složitost rozhraní, pro nové uživatele může prostředí Blenderu působit nepřátelsky a snadno je zahltit množstvím nabízených možností. Také komplexnější scény v Blenderu vyžadují mnohem víc výkonu než SFM a renderování z Cycles engineu při vysokém nastavení kvality může i na výkonných sestavách trvat několikanásobně déle než v SFM, ovšem dosáhne se i mnohem lepší kvality obrazu.

2.5.4 Unity

Unity je multiplatformní herní engine vyvinutý Unity Technologies. Vybral jsem ho vzhledem k jeho oblíbenosti mezi indie vývojáři a jeho možnosti využívat jednoduše Kinecty při vývoji aplikací.[18] Unity má velkou základnu uživatelů a nepřeborné množství kurzů a tutoriálů a základní licence je zcela zdarma. Unity má také vlastní Asset Store, kde si vývojáři mohou koupit nebo prodávat své vytvořené herní assety.

3 Příprava Mo-cap studia

3.1 Výběr hardwaru

Hardware jsem vybíral podle potřeb pro vypracování své bakalářské práce. Vzhledem k tomu, že jsem chtěl prozkoumat možnost snímání s několika senzory a rozdíly, které z takové konfigurace plynou oproti použití jednoho senzoru, a také rozdíly při použití Kinect senzorů v1 a v2, potřeboval jsem minimálně dva senzory od obou typů. Dále jsem chtěl zkusit provést hardwarovou úpravu Kinect senzoru v2 pro přímé připojení k PC bez použití adaptéru, takže jsem sehnal ještě třetí senzor typu v2. Tento extra senzor jsem pořídil pro případ, že by se mi úprava nepovedla, a senzor jsem při ní znehodnotil, abych se nemusel poté zdržovat sháněním nového pro snímání dvěma Kinect senzory v2.



Obrázek 7: Mo-cap studio hardware

Poté jsem potřeboval sehnat adaptéry ke Kinectům v2, jelikož jsem sehnal pouze jejich konzolové verze (oba Kinecty v1 jsem sehnal ve verzi pro Windows, takže adaptéry pro ně byli již součástí balení). Poté jsem potřeboval tři vysoké stativy k fotoaparátům, ty sice nejsou naprosto nezbytně nutné, ovšem práci s rozmisťováním senzorů značně zjednodušují. Alternativa by mohla být rozmístit

senzory na nábytek, vypořadit do výšky pomocí krabic a podobně, což ovšem nedoporučuji.

Jako další jsem potřeboval sehnat motion controllery, protože jsem chtěl prozkoumat možnost vytvoření hybridního Mo-cap studia, a zjistit, jak velké rozdíly to může znamenat. Rozhodl jsem se využít Joy-con ovladače určené pro Nintendo Switch, podporu těchto ovladačů iPiSoft přidal teprve v roce 2019 [19] a podporuje s nimi snímání pohybu dlaní a hlavy. [20] Další důvod pro volbu Joy-con ovladačů je jejich malá velikost oproti ostatním iPi Mocap Studiím 4 podporovaným motion controllerům, takže bude mnohem jednodušší jejich přidělení na tělo a práce s nimi.

Touto konfigurací vytvořím hybridní domácí Mo-cap studio, které bude postavené na systému jak optického Motion Capture s Kinect senzory, tak neoptického Motion Capture založeného na prostorových datech získaných z Joy-conů.

Chtěl jsem také provést úpravu jednoho Kinect senzoru v2 a abych to mohl provést, potřeboval jsem k němu vlastní napájecí zdroj a USB-B - USB-A kabel s propustností alespoň 5GB/s. Neberu zde v potaz pájecí stanici a další nástroje potřebné k provedení této úpravy, protože nepředpokládám, že by si je někdo pořizoval jen kvůli úpravě Kinectu. Pokud člověk doma toto vybavení nemá, vyjde mnohem levněji prostě si pořídit adaptér.

Při sestavování studia a rozmisťování senzorů jsem ještě zjistil, že bude potřeba několik prodlužovacích USB kabelů.

3.1.1 Cena vytvoření domácího Mo-cap studia

	Počet kusů	Cena
Kinect v1	2	1 999 Kč
Kinect v2	3	2 500 Kč
Motion Controller	4	1 790 Kč (pár)
Kinect v2 PC adaptér	3	899 Kč
Stativ	3	699 Kč 1 159 Kč 1 500 Kč
Napájecí adaptér	2	189 Kč 439 Kč
Prodlužovací datový kabel USB 3.0	4	293 Kč
USB 3.0 USB-B na USB-A kabel	1	179 Kč
Konektor na propojení k napájecímu zdroji	1	39 Kč
Licence iPi Soft Mocap Studio 4 Basic	1	695 \$
Celkem		39 026 Kč

Tabulka 3: Cena vytvoření mého domácího Mo-cap studia

Cena vytvoření mého Mo-cap studia je okolo 39 000 Kč, ovšem několik položek bylo nutných jen kvůli tomu, že jsem je potřeboval pro vypracování této bakalářské práce. Normálně bych potřeboval jen jeden typ Kinect senzorů, motion controllery nejsou nezbytně nutné a dalo by se ušetřit i na dalších položkách.

Jako své osobní doporučení pro kvalitní snímání bych doporučil následující konfiguraci, pokud je v plánu snímání pouze jednoho herce a uživatel si nechce připlácet za Pro licenci iPi Mocap Studia 4, aby mohl použít konfiguraci více než dvou senzorů.

	Počet kusů	Cena
Kinect v2	2	2 500 Kč
Motion Controller	2	1 790 Kč (za pár)
Kinect v2 PC adaptér	2	899 Kč
Stativ	2	699 Kč
Licence iPi Soft Mocap Studio 4 Basic	1	695 \$
Celkem		25 866 Kč

Tabulka 4: Cena mnou doporučené konfigurace domácího Mo-cap studia

Cena mé doporučené konfigurace se pohybuje okolo 25 000 Kč, využívá dva Kinect senzory v2, s kterými se dá dosáhnout dobrých výsledků snímání, a dvou motion controllerů, snímajících rotaci dlaní a počítá s umístěním obou Kinect senzorů na stativ. Kinect senzory v2 se dají zespoda namontovat na klasický stativ určený pro fotoaparát, na rozdíl od Kinectů v1.

Pokud chce uživatel ušetřit co nejvíc, může cenu ještě rapidně snížit za cenu zhoršení kvality snímání. Pro nejlevnější řešení je možné si vystačit jen s jedním Kinect senzorem v1, vhodně umístěným na nábytku a licencí na základní express verzi iPi Mocap studia 4. Ušetřit se dá i za adaptéry, pokud se povede sehnat verze Kinect v1 pro Windows.

	Počet kusů	Cena
Kinect v1	1	1 999 Kč
Licence iPi Soft Mocap Studio 4 Express	1	199 \$
Celkem		6 547 Kč

Tabulka 5: Cena minimální konfigurace domácího Kinect Mo-cap studia

Tato konfigurace úplně vynechává motion controllery a je čistě optickým Mo-cap studiem. Protože používá jen jeden senzor, je nutné, aby veškeré hlavní body

postavy herce byly neustále viditelné a možnost pohybu herce je tím značně omezená. Tuto konfiguraci doporučuji pouze těm, kteří chtějí co nejlevnější a nejpřimitivnější Mo-cap a nepožadují nijak precizní výsledky.

3.2 Výběr snímacího prostoru

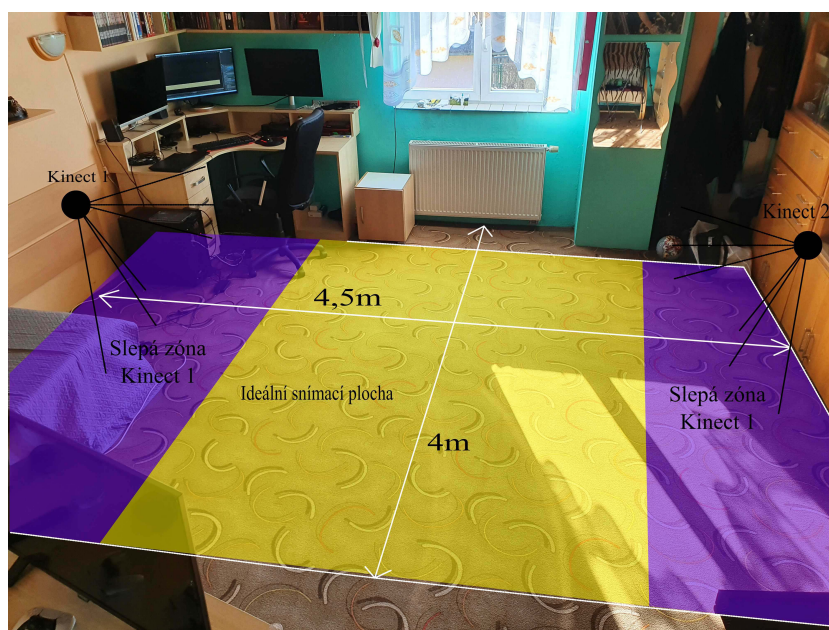
Ideálním prostorem by byla zcela prázdná místnost, kde bych mohl rozmístit senzory ve vhodném úhlu podle jejich počtu pro získání co největší snímané plochy. V domácím prostředí je to s výběrem trochu složitější. Místnost jsem vybral podle velikosti, zaplněnosti, tvaru stěn místnosti (nejvhodnější jsou prosté rovné stěny) a samozřejmě podle využití místnosti, aby její úprava nikomu nevadila.

Zvolil jsem upravit svou pracovnu, která je dostatečně velká (5,5 m x 5 m) a nikdo kromě mě jí moc nevyužívá. Problém byl zaplněnost místnosti a tvar místnosti také není ideální, jedna ze stěn má o 40 cm vystouplý 55 cm široký obdélník kvůli komínu a protilehlá stěna má 20 cm hluboký 100 cm široký výklenek u dveří. Se stěnami dělat nic nešlo, zaplněnost místnosti jsem ovšem vyřešit mohl.

3.3 Příprava místnosti

Začal jsem přesunem velkých kusů nábytku, stolů, gauče a skříní. Tím jsem získal dostatečnou plochu na snímání celé postavy i více senzory, jak lze vidět v nákresu na obrázku č.8. Před přípravou místnosti je potřeba si rozmyslet, kolik senzorů a jak je chci zhruba rozmístit po místnosti. Hloubkové kamery Kinectů v2 fungují zhruba do vzdálenosti od 0,5 m do 4,5 m [1], takže senzory umístěné naproti sobě mi dají průnik cca 3,5 metru aktivní snímací plochy, kterou budou zabírat oba senzory. Kinecty v1 fungují na vzdálenost od 0,4 m až 4 m [1], takže mi dají při umístění naproti sobě cca 3,2 metru průnik snímaný oběma senzory.

Tuto plochu si přesněji změřím a označím, až senzory zapojím a zprovozním záznam z jejich hloubkových kamer, to usnadní práci hercům, aby při snímání vždy věděli, v jaké oblasti se mohou pohybovat.



Obrázek 8: Úprava snímací místnosti - uvolnění snímacího prostoru

3.4 Rozmístění senzorů

Existuje několik doporučených způsobů, jak rozmístit více senzorů pro co největší efektivitu snímání. Pro práci s jedním senzorem ho umístím na jakékoliv vhodné místo, ze kterého bude mít možnost snímat celé tělo herce.

3.4.1 Rozmístění dvou senzorů

Pro snímání dvěma hloubkovými kamerami jsou dvě doporučené možnosti. První je rozložení senzorů, ve kterém jsou v úhlu 60° až 90° v takové vzdálenosti, aby poskytly co největší možný průnik plochy snímání [22], a zároveň měly co nejméně žlutých ploch s vadnými informacemi z hloubkových senzorů (jak je vysvětleno dále).

Druhá možnost je senzory rozmístit téměř úplně naproti sobě v úhlu okolo 180° v takové vzdálenosti, abych měl co největší průnik snímané plochy [22] a zároveň měl co nejméně žlutých ploch s vadnými informacemi z hloubkových senzorů.

Vyzkoušel jsem obě možnosti a podle mě je výhodnější zvolit možnost číslo dvě. Pokud jsou senzory v úhlu 60° až 90° , dosáhl jsem o něco lepších výsledků, co se

týče přesnosti pohybu končetin po všech osách. Nevýhodou bylo, že při otáčení se často stávalo, že zakrytá končetina přestala být snímána - stejně jako při snímání jedním senzorem. Při úhlu 180 ° je snímáno kompletně celé tělo herce z obou stran a je možné se úplně otáčet. Přijde mi, že senzory při této konfiguraci mají trochu problém přesně sledovat pozici končetin, pokud se pohybují směrem od těla a zpět, ale odchylka, která se projevovala, byla maximálně okolo 3 cm. Tento problém by nejspíš vyřešil třetí senzor umístěný ze strany.

3.4.2 Rozmístění tří senzorů

Ideální rozmístění tří senzorů je v rovnostranném trojúhelníku s délkou stran okolo šesti metrů. [22] Získám o něco menší průnik snímací plochy všech tří senzorů než u dvou senzorů ale preciznost snímání při správné kalibraci je vyšší. Toto sestavení eliminovalo problém s nepřesností, který jsem měl při používání dvou senzorů v úhlu 180 °. Při tomto sestavení jsem měl problém sestavit ho tak, abych měl všechny hloubkové senzory bez větších ploch se špatně naměřenými informacemi.

3.4.3 Rozmístění čtyř senzorů

Pro čtyři senzory je ideální rozložení po stranách čtverce, kdy dva senzory jsou vždy naproti sobě. [22] Více hloubkových senzorů iPi Mocap Studio 4 nepodporuje, nejspíš kvůli omezené snímání vzdálenosti hloubkových kamer. Tato konfigurace je doporučena iPiSoftem pro snímání dvou a tří herců, ale vzhledem k velikosti snímání plochy jí osobně nemohu pro tento účel doporučit.

3.4.4 Vliv světla

Vzhledem k tomu, že iPi Mocap Studio 4 při práci s Kinect senzory využívá pro analýzu pohybu herce pouze data z hloubkových kamer [21] a data z RGB kamer jsou pro něj zbytečná, není nutné osvětlení řešit. Snímání by mělo fungovat jak za světla, tak v naprosté tmě, jak si dále v práci ověřím.

Podle mých poznatků ovšem existuje jedna výjimka a to je při kalibraci pomocí světelné značky. Když jsem se o ní pokoušel, musel jsem jí dělat několikrát, pravděpodobně kvůli nevhodnému osvětlení, tím se budu zabývat v podkapitole věnované kalibraci senzorů.

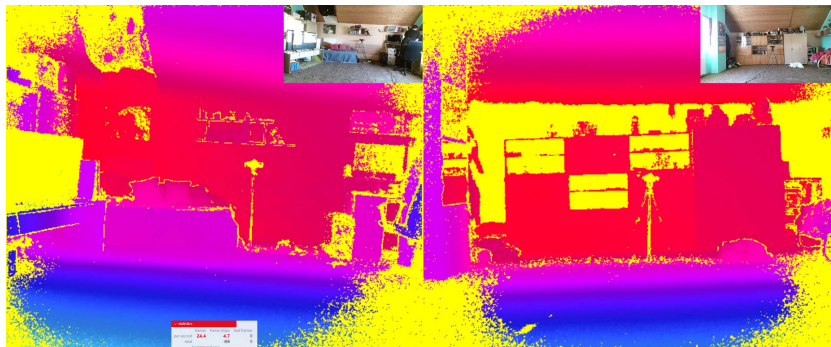
3.4.5 Vliv oblečení herce a pozadí

Další výhodou hloubkových senzorů je, že nemusím řešit barvu oblečení ani pozadí, jako by tomu bylo například u PS Eye kamer, které využívají pro Mo-cap data z RGB kamer. Herec může mít klidně stejnou barvu oblečení jako pozadí za ním a senzor s tím nebude mít při snímání problém.

Pokud bych z nějakého důvodu ovšem data z RGB kamer potřeboval například pro účely klíčování, už bych to řešit musel.

3.4.6 Vliv vzdálenosti

Vzdálenost hraje důležitou roli při použití hloubkových kamer Kinectů. Pokud senzor nevidí ve své limitní vzdálenosti žádnou překážku, vrátí se mu špatná informace, protože nedokázal v daném bodě změřit hloubková data, ta se zobrazují jako žluté plochy ve výstupu z kamery. [19] Stejně tak si hloubkové kamery mohou špatně poradit s odlesky, skleněnými tabulemi a obrazovkami, jak lze vidět na obrázku č.9.



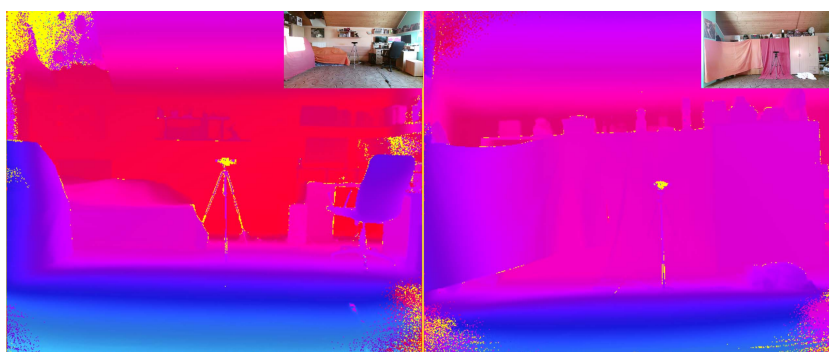
Obrázek 9: Snímek z hloubkových kamer - nezakryté

Je důležité se tyto chybné informace pokusit co nejvíce eliminovat pro dosažení co nejlepších výsledků při snímání, a proto jsem jako další krok naměřil maximální vzdálenost, kam Kinect senzor vidí z hloubkové kamery, a umístil ho tak, abych měl co největší možnou plochu snímání, a zároveň aby jeho hloubková kamera dohlédla na protější stěnu. To samé jsem udělal i s druhým senzorem. Když jsem potom upravoval studio na snímání s Kinecty v1, musel jsem je přiblížit o něco více k sobě, protože dohlédly do menší vzdálenosti než Kinecty v2.



Obrázek 10: Zakrytí problematických ploch

Problém mi dělaly i prosklené skříně a monitory mého počítače, monitor jsem potřeboval, takže s tím jsem nic moc dělat nemohl, ovšem skříně jsem jednoduše překryl prostěradly a to problém vyřešilo. Stejně tak senzory vracely zmatečné informace z oblastí, kde se nacházelo více drobných objektů v malém prostoru kousek od sebe, ty jsem také zamaskoval a získal těmito úpravami výrazně lepší snímací obraz.



Obrázek 11: Snímek z hloubkových kamer - zakryté

Několik žlutých ploch mi bohužel zbylo a nebyl jednoduchý způsob, jak je vyřešit, naštěstí skoro všechny byly v okrajových oblastech, které jsem ke snímání využívat nepotřeboval. Jak vypadal záznam z hloubkových kamer po zakrytí problematických oblastí lze vidět na obrázku č.11

3.4.7 Označení snímané plochy

Dobrym nápadem je označení snímané plochy pro herce, aby se neustále nemusel dívat na monitor, zda-li je stále celý v záběru. Kontrolou aktuálního obrazu z ka-

mer označím jejich viditelnou oblast. Většinou se používají například samolepicí barevné pásky, které se umístí jako značky na podlahy do limitních oblastí snímaného prostoru. V mém případě to kvůli koberci udělat takto nešlo, takže jsem prostor označil jednoduše dalším prostěradlem, které jsem pro různé konfigurace Kinectů mohl jednoduše upravovat jeho přehýbáním a posouváním.



Obrázek 12: Označení snímené plochy

3.5 Zapojení senzorů a motion controllerů

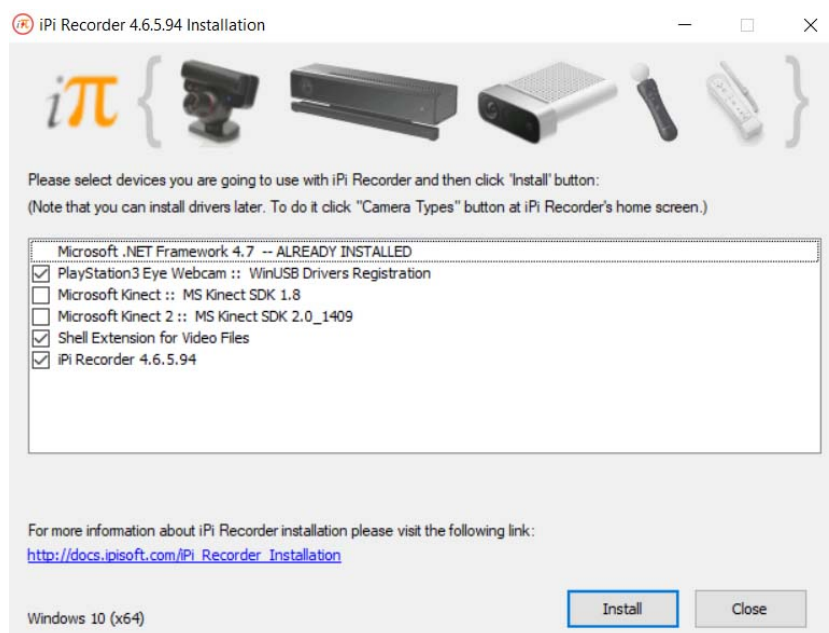
3.5.1 Zapojení Kinect v1 senzorů



Obrázek 13: První zapojení Kinectu v1 - chybějící Kinect SDK 1.8

Zapojení Kinectů v1 je jednoduché, napájení se zapojí do klasické zásuvky a USB kabel do USB portu na počítači. Jedná se o starší typ USB 2.0. a může se zapojit samozřejmě i do USB 3.0 portu.

Pokud je Kinect senzor zapojen do počítače poprvé, správce zařízení v počítači ho rozezná správně jako Kinect kameru, ale nebude mít nainstalovaný vhodný ovladač, a bez něj nebude senzor fungovat. Tento problém jde vyřešit několika způsoby, budu potřebovat The Kinect for Windows Software Development Kit (SDK) verze 1.8, který lze získat přímo ze stránek microsoftu. [23] Tato verze je určená pouze pro Kinect typu v1, pro v2 budu muset nainstalovat jinou sadu.



Obrázek 14: iPi Recorder 4 - instalace potřebných součástí

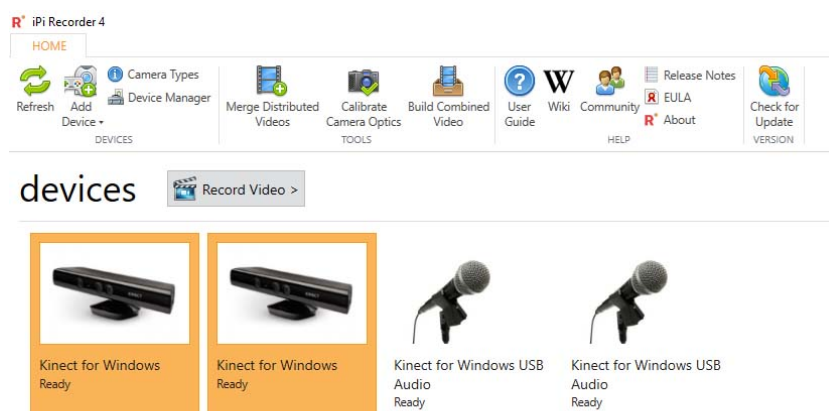
Další možnost je jako první nainstalovat iPi Recorder 4, který při instalaci zkontroluje, jestli v počítači jsou všechny potřebné ovladače a updaty pro práci se zvolenými kamerami, a dá uživateli možnost je nainstalovat (viz. obrázek č.14).



Obrázek 15: Zapojení Kinectu v1 - nainstalované Kinect SDK 1.8

Po instalaci Kinect for Windows SDK 1.8 se Kinecty v1 objeví ve správné kategorii ve správci zařízení a můžou je začít používat. Do jednoho počítače je možné

zapojit několik Kinectů v1 najednou, pokud na ně je v počítači dostatek USB portů, nedoporučuji ovšem používat externí USB huby. Když jsem je zkoušel použít, systém hlásil chybovou zprávu o nedostatku zdrojů pro zařízení při zapojení druhého Kinectu, nebo se zasekl ve smyčce, kdy se zařízení připojovalo a odpojovalo. Pro každý senzor je zřejmě potřeba samostatný port vedoucí do základní desky, protože jeden USB port pravděpodobně nezvládá zpracovat proud dat ze dvou těchto periferních zařízení najednou.



Obrázek 16: iPi Recorder 4 - výběr kamer

V iPi Recorderu by měla nyní při zapojení Kinect senzorů v1 být možnost vybrat senzory ke snímání, jak je vidět na obrázku č.16.

3.5.2 Zapojení jednoho Kinect v2 senzoru

Proces zapojení a zprovoznění Kinectu v2 je téměř identický s Kinectem v1, který jsem popsal v předchozí části práce, proto zde proberu jen ty části, které jsou rozdílné.



Obrázek 17: První zapojení Kinect v2 - bez Kinect for Windows SDK 2.0

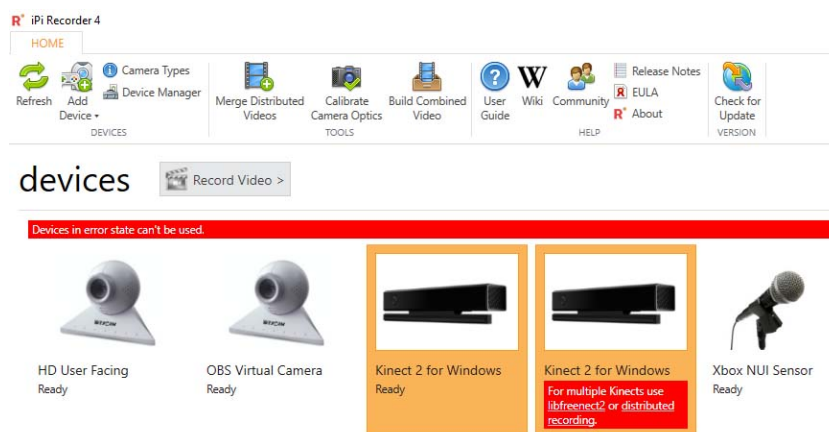


Obrázek 18: Zapojení Kinect v2 - nainstalované Kinect for Windows SDK 2.0

Kinecty v2 vyžadují novější verzi USB 3.0, USB 2.0 porty již nejsou dostačující. Při prvním zapojení je počítač rozeznal jako Xbox NUI senzor, což znamená, že Windows rozpozná, zda se jedná o konzolovou verzi senzoru nebo zda je přímo určená pro Windows. Další rozdíl je, že narozdíl od Kinectu v1 služba Windows update našla sama ovladač pro novější verzi Kinectu. Kinect se sice po instalaci tohoto ovladače zařadil do správné kategorie ve správci zařízení, ale iPi Recorder 4 s ním ani potom nedokázal pracovat. Je tedy potřeba nainstalovat The Kinect for Windows Software Development Kit (SDK) verze 2.0, který lze také stáhnout ze stránek Microsoftu [24], nebo jí nainstalovat jako součást instalace iPi Recorder 4.

Nyní je možné vybrat zapojené Kinect v2 senzor pro snímání v iPi Recorderu.

3.5.3 Zapojení více Kinect v2 senzorů do více počítačů



Obrázek 19: Kinect v2 - pokus o zapojení více senzorů s oficiálním ovladačem do jednoho PC

Kinect senzory v2 s oficiálním ovladačem od Microsoftu podporují použití pouze jednoho senzoru připojeného k počítači. Pokud chci snímat s více kamerami typu

v2 a použít oficiální ovladač, musím mít i více počítačů s USB 3.0 porty a do každého zapojit jeden Kinect v2. Při pokusu o snímání s více Kinecty v2 s oficiálním ovladačem iPiSoft Recorder nahlásí chybu, jak je na obrázku č.19.

3.5.4 Nastavení master-slave v iPi Recorder 4

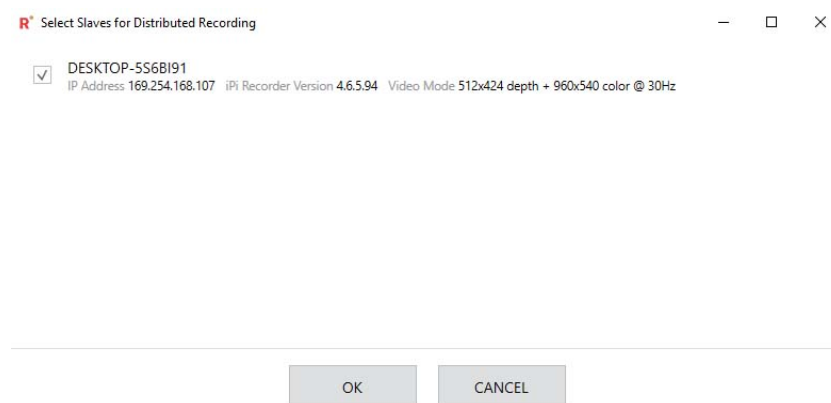


Obrázek 20: Propojení počítačů pro distribuované snímání

iPi Recorder 4 to řeší využitím modelu Master-Slaves, kdy se jeden počítač zvolí jako řídicí, a ostatní jako jeho otroci, které se nechávají ovládat příkazy z řídicího počítače. Je nutné, aby spolu počítače mohly komunikovat, například připojením ke stejné Wi-fi síti nebo propojením kabelem přes ethernet porty, pokud je oba počítače mají, na obou počítačích je potřeba mít nainstalovaný iPi Recorder 4 a Kinect for Windows SDK 2.0.

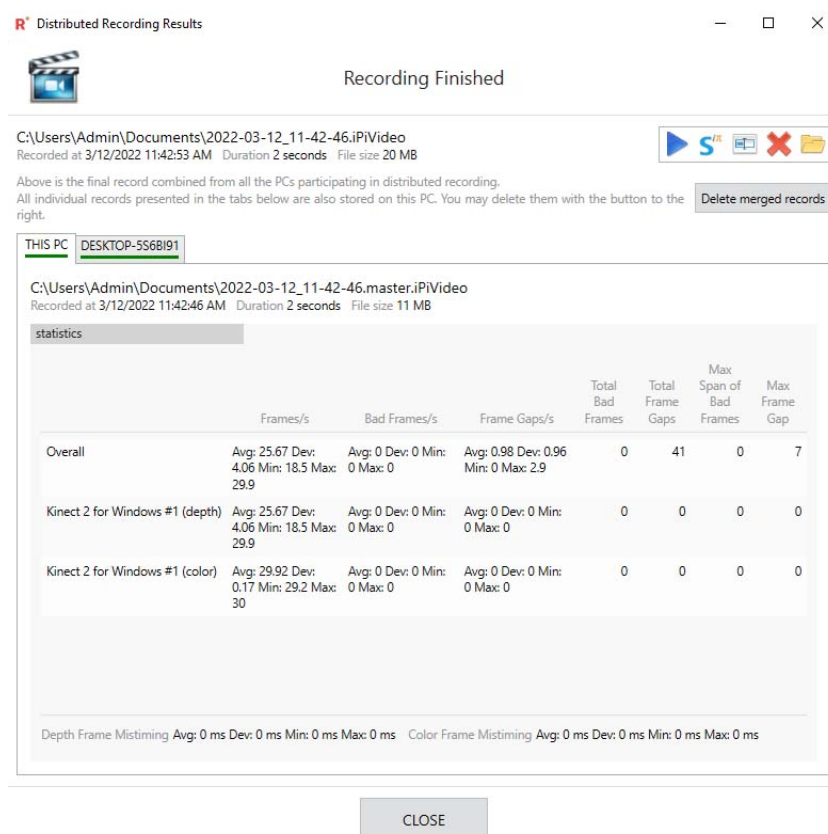


Obrázek 21: iPiRecorder 4 - Master-Slaves nastavení



Obrázek 22: iPiRecorder 4 - výběr otroků

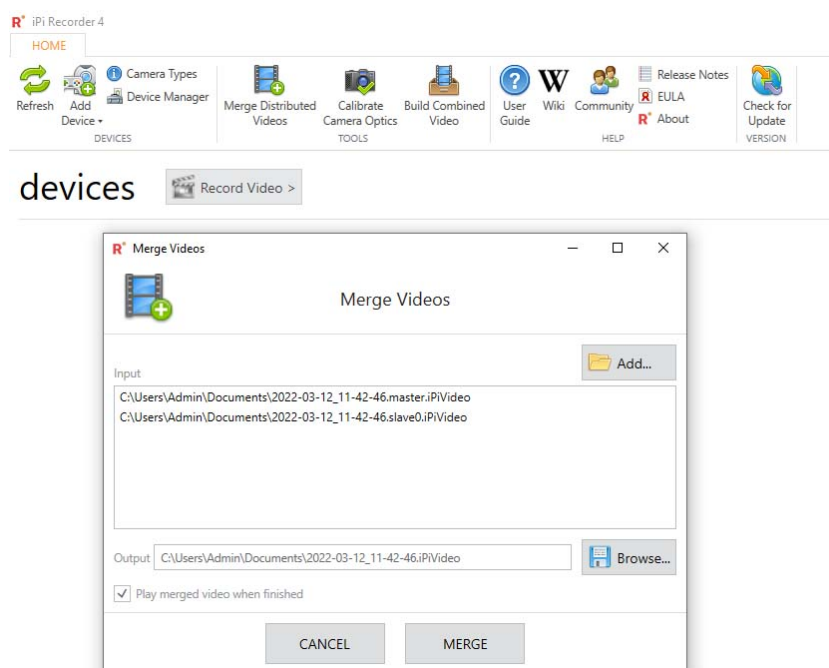
V záložce Record u otroků je potřeba vybrat možnost Enter Slave Mode a nastavit správný síťový adaptér, ke kterému budou připojené všechny počítače, které jsou ke snímání potřeba. V řídicím počítači vyberu možnost Connect to Slaves a objeví se tabulka s nalezenými otroky, můžu si vybrat, které chci při snímání použít. [26]



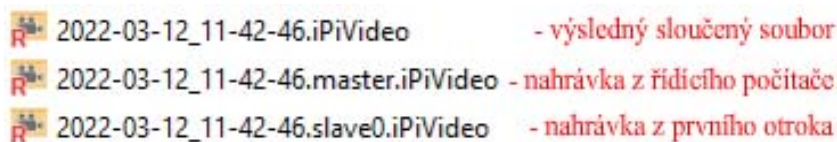
Obrázek 23: Automatické sloučení sekvencí při distribuovaném snímání v iPi Recorderu 4

Když se spustí snímání z řídicího počítače, spustí se i na všech k němu připojených otrocích. Po skončení snímání se uloží na každém počítači jeho část záznamu a pokusí se jí poslat ke sloučení do řídicího počítače. [26] Při použití klasické Wi-Fi bude tento způsob ve většině případů velmi neefektivní vzhledem k velikosti nasnímaných souborů a přenosové rychlosti použité Wi-Fi sítě a navíc se mi několikrát stalo, že se vyskytla při přenosu nějaká chyba a sám se zrušil. Osobně při nahrávání ze tří senzorů v master-slave módu při použití Wi-Fi se minutu dlouhá sekvence posílala téměř 37 minut. Při použití propojení počítačů ethernet kabely mi přes lokální síť trvala stejně dlouhá sekvence poslat pouhých 10 sekund.

3.5.5 Manuální synchronizace nahraných sekvencí



Obrázek 24: Manuální sloučení sekvencí z jednotlivých master a slave iPiVideo souborů



Obrázek 25: Typy iPiVideo souborů

Pokud není možnost zajistit rychlý automatický přenos nasnímaných sekvencí mezi počítači ke sloučení, existuje další možnost, jak udělat sloučení manuálně. IPi Recorder ukládá sekvence ve formátu iPiVideo, při použití distribuovaného snímání má každý soubor v názvu buď označení master nebo slave (jak je ukázáno na obrázku č.25). Tyto soubory lokalizují na všech počítačích a přenesu je například na externí disk do společné složky. V iPi Recorderu na domovské kartě se vybere možnost Merge Distributed Videos a následně je potřeba najít všechny části záznamu

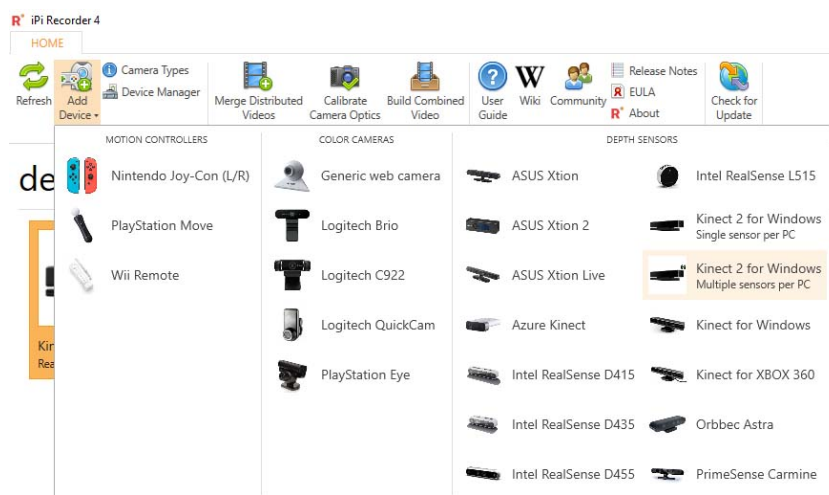
a vybrat možnost Merge, čímž se vytvoří nový iPiVideo soubor se všemi částmi záznamu, který můžu následně zpracovat v iPi Mocap Studiu.

Každý jednotlivý iPiVideo soubor je možné zpracovat v iPi Mocap Studiu i bez sloučení, ovšem bude s ním možné pracovat jen jako se sekvencí snímanou jedním senzorem, který byl připojený k danému otrokovi nebo řídicímu počítači.

3.5.6 Zapojení více Kinect v2 senzorů do jednoho počítače

Existuje možnost, jak zapojit do jednoho počítače více Kinectů v2, a provádět s nimi snímání. Je potřeba použít neoficiální ovladač libfreenect2 a je nutné, aby tento ovladač byl podporovaný zvoleným snímacím softwarem. Dalším požadavkem je dostatečně výkonný počítač, který zvládne zpracovat data z více Kinect v2 senzory najednou, a také musí mít samostatné USB 3.0 PCI-E karty pro správnou funkčnost, jak uvádí vývojáři. [26] Tímto způsobem by mělo být možné do jednoho počítače zapojit a používat až 4 Kinect v2 senzory najednou.

Jsou dvě možnosti, které ovladače Kinectu nainstalovat, abych jim umožnil pracovat s libreenect2, jsou to UsbDk a libusbK. LibusbK driver se musí instalovat přes Zadig a zcela nahrazuje původní Kinect ovladač, tuto možnost je vhodné využít pouze pokud z nějakého důvodu UsbDk způsobuje problémy s dalšími USB zařízeními.



Obrázek 26: iPi Recorder - více v2 senzorů na jednom PC s využitím libfreenect2

Po instalaci ovladače UsbDk v iPi Recorderu vyberu možnost Add Device a přidám Kinect 2 for Windows Multiple Sensors per PC. [27] Nyní je možné na počítači použít více Kinectů v2 pro snímání, pokud na to má dostatečný výkon. Na mé sestavě (Intel Core i9-7960X, Nvidia GeForce RTX 3060ti a Nvidia GeForce RTX 2070, 32GB RAM DDR5) mi fungovaly všechny 3 senzory v2 najednou v pořádku, i když občas se projevil menší pokles ve snímcích za vteřinu u jednoho nebo dvou senzorů. Tento pokles se vyskytoval jen při užívání některých USB 3.0 portů, po odpojení ostatních USB zařízení kromě klávesnice a myši všechny fungovaly stabilně při 30ti snímcích za vteřinu.

3.5.7 Zapojení do jednoho počítače vs. Master-Slaves mode

Dle mého názoru je využití libfreenect2 pro zapojení více senzorů do jednoho počítače nejlepší variantou, pokud to počítač výkonově zvládne. Master-slave varianta má příliš mnoho nevýhod, jako potřeba vlastnit více počítačů na připojení Kinectů, což značně může zvýšit cenu Mo-cap studia, a také problémy s rozmístěním kabelů a počítačů po snímací místnosti, problematické je i propojení počítačů, kdy mi při propojení přes Wi-Fi i ethernet kabel občas počítače prostě hodily chybu a odpojily se ze slave módu přímo při nahrávání. Nepříjemnou překážkou může být i pomalý přenos nahraných sekvencí z otroků do řídicího počítače a jejich spojení.

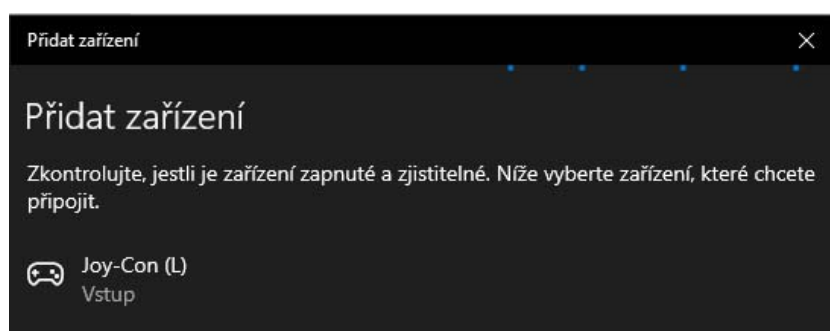
V tomhle porovnání je dle mého názoru jednoznačným vítězem využití libfreenect2.

3.5.8 Spárování motion controllerů

Jako motion controlery jsem si vybral Joy-cony, kvůli jejich kompaktní velikosti a jejich jednoduchému propojení s Windows 10. Potřeba je jen, aby počítač umožňoval připojení Bluetooth zařízení, pokud to neumí, lze zakoupit i USB Bluetooth adaptér.



Obrázek 27: Joy-con v režimu párování



Obrázek 28: Windows 10 - Párování Joy-conu přes Bluetooth

Na Joy-conu podržím pár vteřin černé tlačítko nacházející se na části, která slouží pro přichycení Joy-conu (viz. obrázek č.27) ke konzoli Nintendo Switch. Až na něm začnou blikat zelená světla, v počítači půjdu do nastavení počítače a do kategorie Nastavení Bluetooth a Dalšíh Zařízení, kde stisknu Přidat Zařízení Bluetooth nebo Jiné. V dalším okně, které se objeví, vyberu možnost Bluetooth, a až se objeví můj Joy-con, kliknu na něj. Joy-con by se měl objevit v seznamu spárovaných zařízení (jako na obrázku č.29), takto spáruji všechny Joy-cony, které hodlám používat. V iPi Recorderu je nyní můžu vybrat pro použití při snímání.

Zařízení Bluetooth a jiná



Přidat zařízení Bluetooth nebo jiné

Bluetooth



Zapnuto

Vaše zařízení se teď dá zjistit jako DESKTOP-NCU601L.

Myš, klávesnice a pero



Joy-Con (L)

Připojeno

Obrázek 29: Joy-con spárovaný s Windows 10

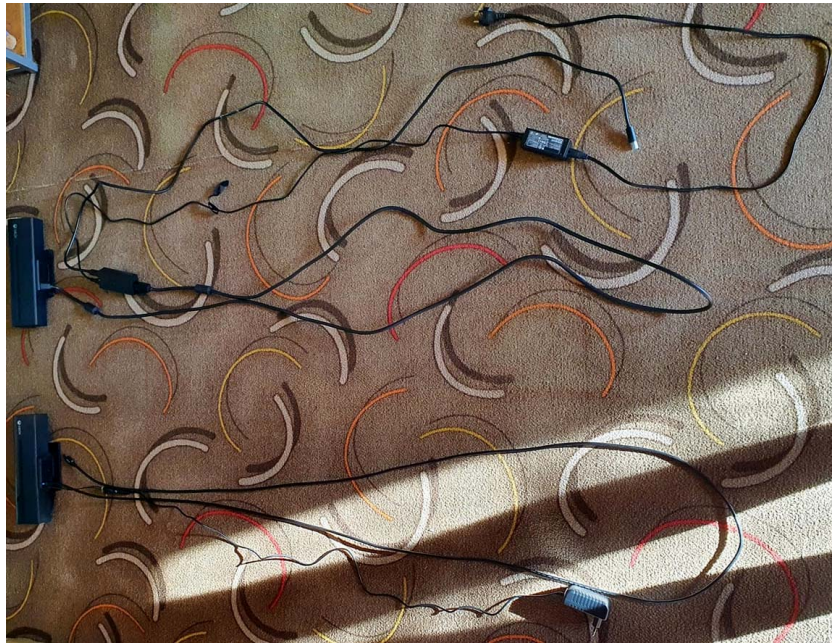
4 HW úprava Kinectu v2

Existuje možnost, jak konzolovou verzi Kinectu připojit k počítači, je ovšem potřeba ho rozebrat a provést menší hardwarovou úpravu. Tato možnost mi přišla zajímavá, a proto jsem se jí rozhodl v rámci této práce prozkoumat a zjistit, jestli stojí za to se jí zabývat a ušetřit nějaké peníze na pořízení adaptéru.

4.1 Důvod úpravy Kinect senzoru

Je několik důvodů, proč by uživatel mohl chtít svůj Kinect senzor upravit tímto způsobem.

1. **Finanční** - Pokud již uživatel vlastní potřebné nástroje, náklady na tuto úpravu vyjdou na zhruba 300 - 500 Kč, podle zvolených druhů kabelů a napájecího zdroje. Je možné, že i potřebné nevyužité kabely najde uživatel u sebe doma a náklady mohou být prakticky téměř nulové. Oproti tomu, ceny adaptérů se pohybují okolo 900 - 1500 Kč a v některých obchodech pomalu narůstají, jak se zmenšuje jejich dostupnost.
2. **Dostupnost adaptérů** - Microsoft přestal oficiální adaptéry vyrábět již před několika lety [28] a jejich dostupnost se začíná pomalu zmenšovat. Dnes se ještě dají najít poměrně snadno, ale ceny jdou pomalu nahoru a v obchodech pomalu dochází. Je možné, že tato navrhovaná možnost se za pár let stane tou nejsnadnější možností, jak Kinect v2 k počítači připojit.
3. **Praktičnost** - Kabel pro zapojení Kinect senzoru do konzole je sám o sobě dlouhý zhruba 3 metry a celková délka adaptéru je okolo okolo 2,5 metru a skládá se z několika kabelů a dvou propojovacích boxů (jeden na rozdělení napájení a datového kabelu a druhý kvůli napájení). To je poměrně hodně kabelů, které může být problém správně rozmístit a schovat tak, aby se nepletly pod nohy, obzvlášť pokud je v plánu použít senzorů víc než jeden.



Obrázek 30: Porovnání délky kabeláže upraveného Kinectu v2 a oficiálního adaptéru

Někdo by mohl říct, že dalším důvodem by mohla být i estetika, pokud se pro senzor na 3D tiskárně vytiskne nový zadní panel a zasadí se do něj přímo napájecí koncovka, bezesporu bude toto řešení vypadat lépe, než sebou tahat několik metrů kabelů z adaptéru. Jak je vidět na obrázku č.30, můžu dosáhnout i o několik metrů méně potřebné kabeláže, bylo by možné vybrat i kratší datový kabel než je na obrázku, já zvolil delší, protože potřebuji Kinecty zapojit přes celou místnost.

Tato úprava je také vhodná pro připojení senzoru do novějšího typu konzole Xbox, takže i konzoloví hráči z ní mohou těžit tím, že nebudou mít u televize hromadu zbytečné kabeláže navíc.

4.2 Jak úprava funguje

Tato úprava je neoficiální a jejím provedením uživatel ztrácí záruku. Na internetu se nevyskytuje žádný oficiální návod na to, jak jí provést ani přesná specifikace, jak doslova funguje. Proto informace v této kapitole jsou buď kousky, co jsem posbíral po různých fórech, a i když si myslím, že jsou správné, nejde je brát jako 100% ověřené. Něco jsem zjistil sám při své práci. V tom případě se jedná o mé domněnky

a opět netvrdím, že jsou 100% přesné a neměly by být brány jako ověřená fakta.

Při svém pokusu o úpravu Kinect senzoru jsem vycházel především z informací, které mi poskytl tutoriál umístěný na webovou stránku <https://imgur.com/> od uživatele redditu pod přezdívkou Eagle115. [29]

Z toho co jsem pochopil, Kinect v2 vyžaduje dodatečné napájení 12V, bez kterého nemůže fungovat, to Microsoft vyřešil vytvořením speciálního kabelu, který v sobě kombinuje jako možnost přenášet data, tak si toto přídatné napájení vzít přímo z konzole Xbox. Jedná se vlastně o upravený kabel typu USB-B s druhou koncovkou do konzole Xbox.



Obrázek 31: Klasické USB-B 3.0 vs. originální Kinect v2 kabel

Při bližším prozkoumání je zřejmé, že v tomto kabelu jsou extra piny po stranách koncovky kabelu, jak lze vidět na obrázku č.31, oproti klasickému USB-B, které se pravděpodobně starají o dodatečné napájení. Podle této teorie, by mělo být možné oddělit dodatečné napájení a datový přenos, o které se starají datové

piny v USB-B kabelu.

Napadají mě dvě možnosti, jak to provést. První možnost by mohla být rozříznout a upravit samotný kabel vedoucí z Kinectu, spájet správně datové piny ke klasickému USB-B na USB-A kabelu standartu 3.0 a napájecí dráty připájet k externímu zdroji napájení. Co je na této možnosti lákavé, je určitě to, že není potřeba vůbec zasahovat do samotného zařízení Kinectu. Já tuto verzi ale zavrhl, protože mi přišla velmi složitá. Nebyl jsem si jistý, jak od sebe rozeznat páry datových kabelů, které k sobě spájet a USB 3.0 vyžaduje určitý datový tok, který bych velmi pravděpodobně tímto procesem narušil.

Druhou možností je původní kabel zcela vyřadit a na piny na základní desce Kinectu, které se starají o přídatné napájení, připájet přímo externí napájecí zdroj a do Kinectu poté zapojit klasický USB 3.0 USB-B - USB-A kabel s propustností alespoň 5Gb/s, pokud by byla nižší, Kinect by nemohl správně fungovat.

Výsledkem této metody získám konzolový Kinect v2, který bude možné pomocí klasického kabelu, kterým se někdy připojují k počítači například tiskárny a napájecího zdroje z jiného zařízení (možné je využít třeba starší nabíječku k telefonu atd...) mu zajistit potřebné dodatečné napájení, které klasický USB port nezvládne.

4.3 Rizika navrhované úpravy

Je důležité si dobře rozmyslet, jestli se do této úpravy pustit, na straně uživatele je vyžadována určitá zručnost a znalosti ohledně pájení a elektroniky celkově.

Existuje samozřejmě riziko, že se senzor poškodí, a to především při nesprávném připájení napájecího zdroje k desce Kinectu, ale stejně tak je možné ho poškodit při rozebrání a neodborné manipulaci. Piny na desce Kinectu v oblasti pájení jsou velice blízko u sebe a je tam málo prostoru pro nepřesnost, je poměrně velké riziko, že se omylem povede piny přemostit a zařízení vyzkratovat.

Riziko zranění hrozí i uživateli, který by neodborně zacházel s potřebnými nástroji. Je nutná práce s pájkou, která je velice horká, takže hrozí riziko popálení a výpary pájecího materiálu mohou být toxické. Uživatel by měl užít ochranných brýlí a vhodnou ochranu dýchacích cest a o úpravu by se neměly v žádném případě pokoušet malé děti, ani dospívající bez dohledu dospělé osoby.

4.4 Potřebné komponenty a nástroje

K provedení úpravy Kinectu jsem využil následující komponenty a nástroje.

1. Kinect v2
2. Napájecí zdroj Evolveo 12V/2A
3. Konektor k propojení napájecího zdroje a drátů připájených ke Kinectu
4. Kabel USB-B - USB-A 3.0
5. Prodlužovací kabel k napájecímu zdroji
6. Multimetr Retlux RDM 9001
7. Pájecí stanice Extol Industrial
8. Sada šroubováků VOREL
9. Horkovzdušná pistole WAGNER

Každé položce a důvodu jejího výběru a využití se podrobně věnuji ve video tutoriálu, který jsem vytvořil, a je umístěn na youtube a mnou vytvořeném webu ohledně Mo-cap s Kinecty howtomocap.com. V tomto tutoriálu se věnuji velmi detailně popisu celému procesu úpravy Kinectu a vřele doporučuji jeho zhlédnutí pro více informací.



Obrázek 32: Nástroje a komponenty potřebné pro úpravu Kinectu

4.5 Proces úpravy Kinectu



Obrázek 33: Kinect v2 s vytaženým kabelem

Nejprve je potřeba z Kinectu odpojit jeho originální kabel, může se zdát, že je tam napevno, ale je normálně odpojitelný, má u sebe gumové těsnění, která ho drží na místě, a je potřeba vyvinout poměrně dost síly na jeho odpojení. Je důležité tahat rovně a ne do stran, abych neponičil konektor uvnitř Kinectu, kam budu poté zapojovat nový klasický USB-B kabel a sílu zvyšovat postupně a pomalu, nesnažit se za kabel trhat. Tento kabel již nadále není potřeba a je vhodné si ho někde rovnou uklidit.



Obrázek 34: Nahřátí polepů pro snadné odstranění

Kinect je nyní potřeba otočit a zbavit se nálepek z jeho spodní strany, které brání v přístupu ke šroubkům. Je samozřejmě možné je seškrábat a otrhat. Já jsem je nechtěl zničit, abych je mohl později vrátit, a hlavně jsem nechtěl Kinect poškrábat, proto jsem využil horkovzdušnou pistoli a nálepky si nejprve zahřál (viz. obrázek č.34, poté jsem je odstranil velmi snadno a bez poškození. Je potřeba dát velký pozor, aby se senzor moc nepřehřál, horkovzdušná pistole ho dokáže rozpálit až na několik stovek stupňů Celsia a mohl by se poškodit. Alternativou je použití klasického fěnu, který zde také odvede svou práci, i když pomaleji.



Obrázek 35: Klasický Torx 10 vs. bezpečnostní Torx 10



Obrázek 36: Odšroubování bezpečnostního Torx pomocí malého plochého šroubováku

Pod nálepkami se nachází na každé straně 4 šroubky, které je potřeba odšroubovat, 4 z nich jsou klasické Torx 10 šroubky a 4 jsou bezpečnostní Torx 10. Ty se liší tím, že uprostřed mají malou kuličku (viz. obrázek č.35), která brání jejich odšroubování, a také jsou delší. To jsem netušil a měl jsem s tím menší problém, protože v mé sadě jsem měl sice klasický Torx 10, ale bezpečnostní Torx 10 mi chyběl. Při pokusech o jeho odšroubování jsem zkusil kuličku uprostřed zbrousit nebo vylomit, což nepřineslo příliš dobré výsledky. Ovšem nakonec se mi podařilo je odšroubovat jednoduše použitím nejmenšího plochého šroubováku a jeho vložením mezi bezpečnostní prvek uprostřed hlavy šroubku a jednu z jeho stěn, pod úhlem několika stupňů se mi ho tam podařilo zaseknout a následně je odšroubovat, jak ukazuje obrázek č.36. Tento proces můžete detailně vidět v již výše zmíněném tutorial videu.



Obrázek 37: Rozdílné délky Bezpečnostních Torx 10 (nalevo) a torx 10 (napravo)

Z Kinectu jsem nyní sundal spodní část jeho konstrukce pomalým pohybem vzhůru, měla by klást pouze zcela minimální odpor a rozhodně na ní není možné působit silou, jinak se úchyty poškodí. Poté jsem odstranil obě bočnice ze stran Kinectu, ty na místě nedrží žádné šrouby, je potřeba pouze opatrně zatlačit na vrchní část Kinectu a tím je uvolnit a následně je druhou rukou vyndat. Je nutné dát si pozor, aby nedošlo k ohnutí žádné z tenkých plechových součástí, které se v okolí bočnic nachází. Odebráním bočnic se mi odhalily další dva klasické Torx 10 šroubky (viz. obrázek č.38), které držely horní a přední část plastové konstrukce Kinectu.



Obrázek 38: Odebrání dvou skrytých Torx 10 šroubků držících přední a vrchní část krytu Kinectu

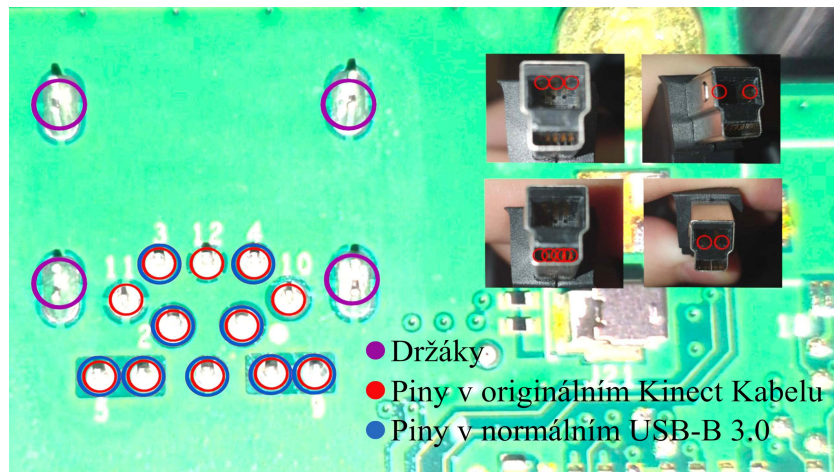
Sundání dílu konstrukce, který kryje přední a vrchní stranu senzoru, je o něco ošemetnější, je potřeba vrchní část trochu nadzvednout a mírně zatlačit dopředu, poté chytnout spodní okraj a potáhnout ho směrem dolů a vrchní část přitom tlačít lehce vpřed.



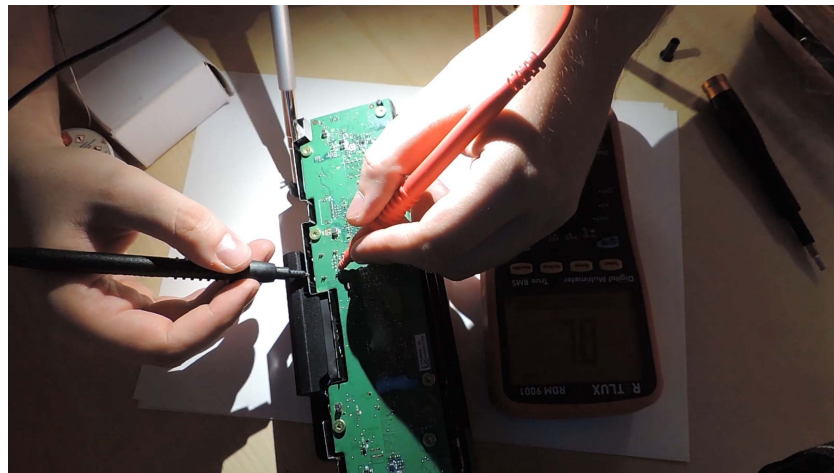
Obrázek 39: Rozložený Kinect v2

Pokud bych chtěl vytisknout kryt pro ventilátor se zabudovaným portem pro napájecí zdroj do těla Kinectu, sundal bych ještě část kryjící ventilátor a vyměnil ji, bohužel 3D tiskárnu nevlastním a neměl jsem možnost si to momentálně zařídit, takže dále nemělo smysl Kinect rozebírat.

Jediná část základní desky Kinectu, která mě v rámci této úpravy zajímala, je část přímo nad konektorem pro kabel, který jsem hned na začátku úpravy odpojil.



Obrázek 40: Hledání pinů vylučovací metodou

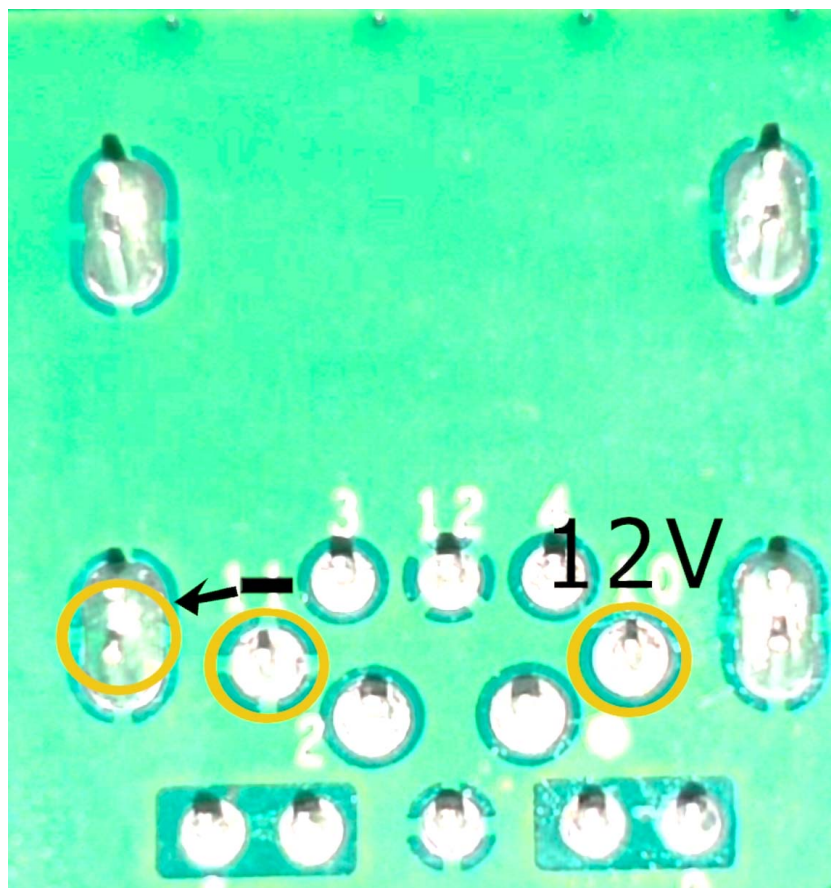


Obrázek 41: Hledání správných pinů pomocí multimetru

Teď jsem potřeboval zjistit, ke kterým pinům bude potřeba připájet kabely pro napájecí zdroj. Hned jsem si všiml, že jsou piny ve stejném uskupení, jako se nachází i v kabelu pro Kinect, a při porovnání s klasickým USB-B, které obsahuje pouze piny pro přenos dat, jsem mohl vylučovací metodou zúžit možný výběr. Vylučovací metodou jsem dospěl k pinům 10,11 a 12, což byly postranní piny v originálním Kinect kabelu a prostřední horní pin. Klasické USB-B jich má jen 9, takže mi dávalo i smysl, že ty, které jsou označené jako 10,11 a 12, mi zbyly jako ty nadbytečné podle chybějících pozic pinů v kabelu (viz. obrázek č.40). Hledal

jsem jeden plusový, který je vstupem pro 12 V, a jeden minusový, neboli nulák.

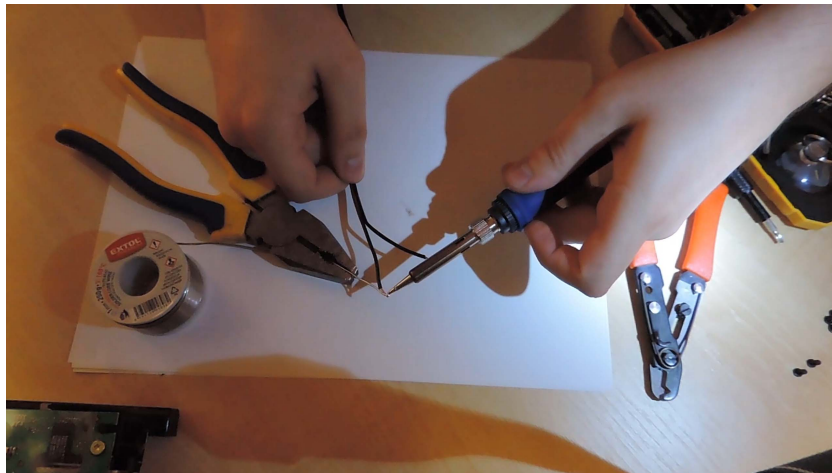
Minusový by měl být propojený s kostrou, potřeboval jsem tedy použít multimetr. Na multimetru jsem nastavil režim testování spojů a měření kapacity a zapnul si bzučák. Pokud se oba konce zapojených měřících kabelů dotknou místa, které je propojené, multimetr vydá zvukový signál, jeden měřák jsem tedy přiložil ke kostře konstrukce a druhý přikládal na jednotlivé piny (viz. obrázek č.41). Piny číslo 10 a 12 nebyly propojené s kostrou, takže mohly být hledaným vstupem pro 12 V a pin 11, byl minusový. K čemu slouží pin číslo 12 netuším a nepodařilo se mi to nikde dohledat, ale vyloučil jsem ho z několika důvodů. Zaprvé se nacházel přímo mezi dvěma piny, které by měly být datové v klasickém USB-B. Nedávalo by podle mě smysl tam umisťovat pin pro přídavné napájení a logicky mi dávalo mnohem větší smysl, že dva protilehlé budou tvořit pár plus a mínus.



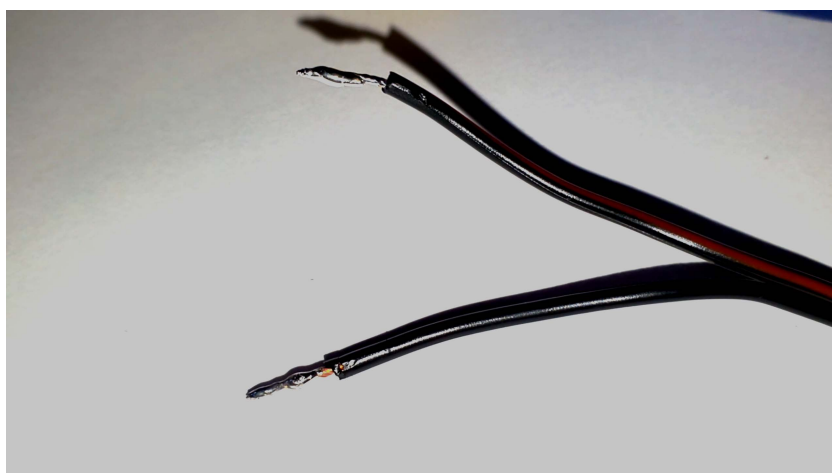
Obrázek 42: Zvolené piny pro připájení kabelů

Jako minusový je možné využít v podstatě jakýkoliv pin, který je propojený s kostrou, a já si zvolil držák vedle pinu 11, protože byl větší a měl kolem sebe více prostoru (viz. obrázek č.42), což znamená menší šanci něco přemostit.

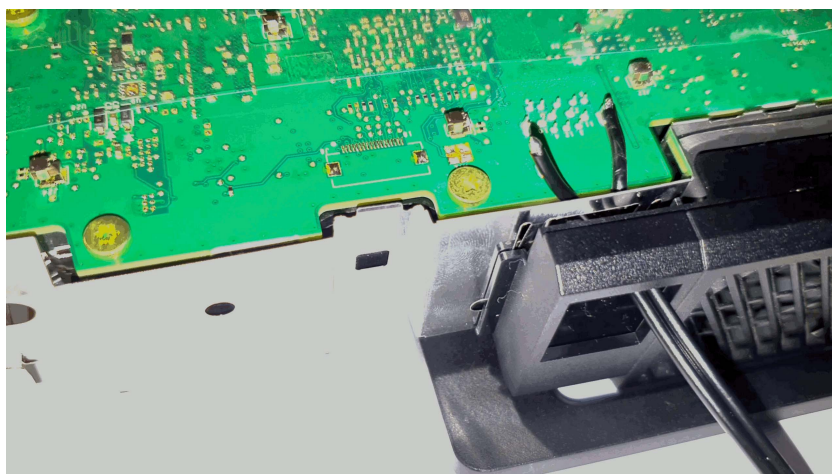
Odstříhl jsem cca 20 centimetrů dlouhý kus prodlužovacího kabelu k napájecímu zdroji, malou část drátu o něco menší než jeden centimetr jsem zcela obnažil, pokud bych na druhém konci měl vhodný konektor pro napájecí zdroj, mohl bych ho tam ponechat. Já takový kabel neměl a musel jsem tedy obnažit oba konce kabelu. Následně jsem dráty na obou koncích lehce pocínoval, což mi mělo usnadnit následné připájení k desce kinectu, a na druhé straně připojení do konektoru umožňující spojení s napájecím zdrojem. Cínování jsme provedl tak, že jsem si zachytil pájecí drát v určité poloze lehce nad povrchem stolu a zapnul pájecí stanicí a nastavil si vhodnou teplotu, po nahřátí pájky jsem přiložil drát přímo pod pájecí drát a přiložil k němu pájku a nechal ho část roztavit a připojit se na drát (viz. obrázek č.43), který jsem chtěl připájet k desce Kinectu. Tento proces jsem opakoval i pro další dráty. Na pocínování stačí velice malé množství pájecího drátu a je nutné dát pozor, aby se uživatel o zapnutou páječku nespálil a nevdechoval výpary z taveného pájecího drátu.



Obrázek 43: Cínování



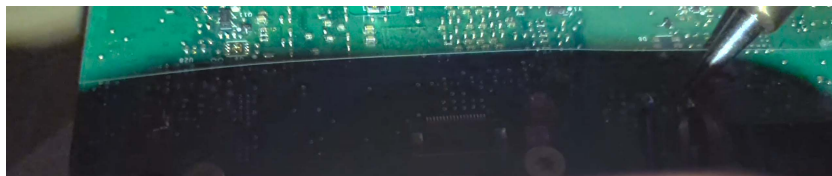
Obrázek 44: Pocínované dráty



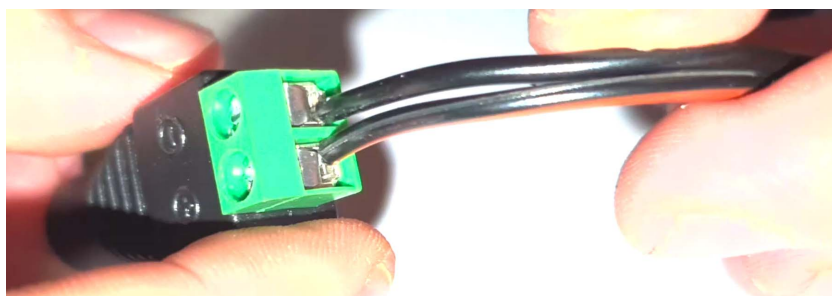
Obrázek 45: Provléčení pocínovaných kabelů za kostřením

Pocínované dráty jsem protáhl konektorem pro připojení kabelu do kinectu hořejškem za kostřením (viz. obrázek č.45). Je nutné je provléci až za kostřením, jinak Kinect nebude moci zase normálně složit. Podle standartu by měl červený drát být v napájecím adaptéru plus a černý mínus, je třeba dát pozor na levné čínské adaptéry, které to mohou mít jinak, a způsobit tím velké problémy. Dráty jsem následně velmi opatrně napozicoval tak, aby se dotýkaly potřebných pinů a ničeho jiného a velmi opatrně je připájel. Při pájení k desce je potřeba být opravdu velmi opatrný, kvůli malému prostoru a množství okolních pinů. Nejprve

je potřeba přiložit pájku k pinu na desce a rozehrát ho, poté přiložit drát a připájet ho k pinu. Není dobrý nápad pájet kabel k nerozehrátému pinu, protože může vzniknout studený spoj, a je také potřeba nepoužít příliš pájecího drátu.

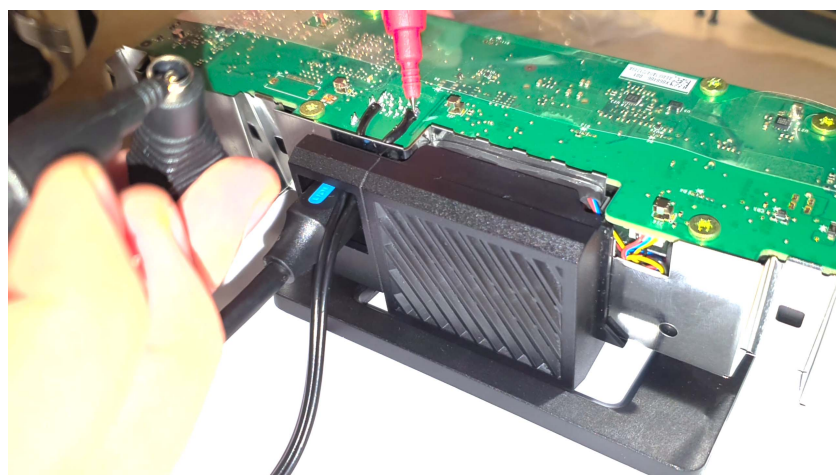


Obrázek 46: Připájení kabelů k základní desce Kinectu



Obrázek 47: Dráty zapojené do konektoru pro připojení napájecího zdroje

Druhou stranu s pocínovanými dráty jsem vložil do konektoru pro propojení s napájecím zdrojem a přišrouboval je tam, je nutné správně připojit plus a mínus. V tento moment je možné Kinect složit a zapojit, ale je lepší provést kontrolu, zda pájení bylo úspěšné. Kontrolu spojení desky s konektorem jsem provedl pomocí multimetru (viz. obrázek č.48), přiložil jsem jeden měřicí drát na stěnu konektoru a druhý na pin, ke kterému jsem připájel mínus, a poté jsem přiložil první kabel na prostředek konektoru a na pin, ke kterému jsem připájel plus, v obou případech multimetr hlásil spojení, takže kontrola byla úspěšná.



Obrázek 48: Kontrola správného propojení konektoru s deskou

Nyní je možné Kinect složit a připojit k počítači. Kroky jsou stejné, jako při jeho rozložení, akorát v opačném pořadí. Do Kinectu jsem zapojil USB 3.0 USB-B-USB-A kabel a druhý konec zapojil do počítače, do konektoru pro napájecí zdroj jsem zapojil napájecí zdroj Evolveo 12 V/2 A a zapojil ho do zásuvky.



Obrázek 49: Upravený Kinect v2

V tuto chvíli by se měl Kinect připojit k počítači, bohužel se ale nepřipojil. Problém byl nejspíš v USB kabelu, protože po jeho výměně Kinect začal na připojení k počítači reagovat. Stále ale nefungoval správně, protože se v intervalu několika vteřin připojoval a odpojoval, a při pokusu o snímání měl velmi nízké

snímky za vteřinu. To vyřešilo odpojení dalších nepotřebných periférií od počítače. Domnívám se, že problém byl v tom, jak je Kinect náročnou periférií, že počítač nezvládal zpracovávat data a přiřazovat zdroje všem najednou.

4.5.1 Možné problémy připojení Kinectu po úpravě k počítači

Jak je zmíněno v předchozím odstavci, mé zprovoznění upraveného Kinectu mělo jisté problémy. Dovedu si představit, že by spousta uživatelů v tuto chvíli začala panikařit s tím, že Kinect zničili ale nemusí tomu tak být. Existuje několik možností, proč Kinect nefunguje, i když uživatel provedl úpravu správně.

1. **Nevhodný napájecí zdroj** - Zvolený napájecí zdroj by měl mít alespoň 2 A, je třeba si dát pozor na laciné nekvalitní zdroje z Číny, které mohou mít na štítku sice výkon 3 A, ale ve skutečnosti mají například pouze 1 A. V tu chvíli Kinect nebude fungovat správně. Je možné si proměřit, zda-li zdroj skutečně dosahuje hodnot vypsanych na štítku, ale pravděpodobně se tomu lze vyhnout koupí kvalitního zdroje. Pokud není zdroj připájený přímo na desku, lze ho jednoduše vyměnit.
2. **Nedostatečná propustnost USB kabelu** - Tato možnost je bohužel něco, co příliš uživatel nemůže ovlivnit. Některé kabely s garantovanou propustností 5Gb/s jí nejspíš nedosahují a s takovým kabelem pak Kinect nefunguje. Někdy je vhodné vyzkoušet více USB kabelů, v případě že se Kinect nechce připojit.
3. **Příliš mnoho periférií připojených k počítači** - Je možné, že periferie připojené k počítači vyžadují již tolik zdrojů a proudí přes ně takové množství dat, že počítač už nezvládne zpracovat data z Kinectu a USB controller mu nemůže přiřadit potřebné zdroje. Nejjednodušší řešení je zkrátka veškeré nepotřebné periferie odpojit a tuto variantu tím buď potvrdit nebo vyloučit.
4. **Špatné připájení** - Pokud uživatel neprovedl kontrolu spojení konektoru s deskou po pájení, je možné, že je problém tam. Pokud předchozí možnosti nepomohly Kinect zprovoznit, je potřeba ho otevřít a zkontrolovat multimetrem spojení konektoru se základní deskou Kinectu.

5. **Jiné poškození** - Pokud kontrola propojení konektoru s deskou hlásí, že jsou oba kabely připájené správně, problém je již detekovat těžší. Možná došlo k přemostění některých pinů při pájení a došlo ke zkratu, ale mohou za tím být i jiné důvody. V tuto chvíli je pravděpodobně potřeba začít proměřovat multimetrem, jestli na desce Kinectu funguje vše, jak má, a snažit se najít, kde je přesně problém. To bude pro většinu uživatelů pravděpodobně příliš těžké, a i když by se problém podařilo na desce lokalizovat, jeho oprava nemusí být vůbec jednoduchá, může se ukázat i jako nemožná. Dalším rozbořením možných závad bych se začal odchylovat příliš daleko od toho, co mám v rámci této práce prozkoumat, takže se tématu možných závad na základních deskách Kinectů a možnostem jejich řešení nadále věnovat nebudu.

4.5.2 Proč nepájet napájecí zdroj přímo na desku

Napájecí zdroj je možné napájet přímo na desku kinectu a ulehčit si tím trochu práci s vymyšlením jeho propojení. Pokud se ale zdroj ukáže jako nedostatečný, nebo se časem rozbije, nelze jej jednoduše vyměnit za jiný. Protože se nedá odpojit, je také problematické Kinect přenášet a je lehké spojení zdroje s Kinectem při manipulaci s ním poškodit.

Proto za mnohem lepší variantu považuji na desku připájet dráty s konektorem, do kterého se napájecí zdroj zapojí, pokud bych to neudělal, nemohl bych zkusit vyměnit zdroj za jiný, když se mi Kinect nechtěl připojit k počítači, a přidělal bych si ve výsledku hodně práce navíc.

Ani mnou zvolená varianta není tou nejlepší, z Kinectu vede několik centimetrů kabelu na připojení napájecího zdroje. Zcela nejlepší variantou je podle mě vytisknutí nového krytu na 3D tiskárně, do kterého konektor na připojení napájecího zdroje bude přímo zabudovaný.

4.6 Zhodnocení vhodnosti úpravy Kinectu

Úprava Kinectu sice splnila svůj účel a ušetřila nějaké peníze, které by stál oficiální adaptér od Microsoftu, ale když proti sobě porovnám všechna pro a proti, nemůžu její provedení doporučit nikomu, kromě malého okruhu lidí, kteří mají zkušenosti s pájením, rozumí dobře elektronice, mají již veškeré potřebné vybavení a nevadí

jim riskovat zničení senzoru a ztráta záruky.

Přijde mi, že ušetření pár stovek korun nestojí za risk zničení senzoru za několik tisíc, a pájení na desku Kinectu vyžaduje poměrně vysokou přesnost, takže je jednoduché při tom udělat chybu, která může senzor úplně zničit. Dále vzhledem ke všem dalším faktorům, které mohou zprovoznění senzoru komplikovat, si myslím, že dokud je poměrně snadná dostupnost oficiálních adaptérů, je jejich využití tou lepší volbou.

5 Snímání sekvencí

Teď, když je Mo-cap studio sestavené a vše ostatní připravené, je čas se pustit do samotného snímání. V této části práce hodlám popsat přesný postup, jak dosáhnout co nejlepšího snímání s různými konfiguracemi zapojených Kinectů a hlavně porovnat a zjistit kvalitu snímání a vhodnost použití jednotlivých konfigurací.

5.1 Získání dat pro porovnání s vypovídající hodnotou

Je jasné, že proti sobě nelze jen tak náhodně porovnávat jablka a hrušky, proto jsem se pečlivě zamyslel nad tím, jak ke snímání sekvencí pro porovnání přistupovat a jakých pravidel se musím držet, aby mé závěry byly relevantní a měly vypovídající hodnotu. Je také nutné si uvědomit, že zpracování dat z Kinectů je také závislé na využitém softwaru, takže závěry, ke kterým zde dojdu, nemusí být zcela shodné při využití jiného softwaru, jako je například Brekel.

Podle mě jsou tři nejdůležitější body, které je nutné dodržet.

1. **Porovnávat pouze stejný typ pohybu nasnímaný za stejných podmínek;** Bude určitě velký rozdíl mezi kvalitou nasnímaní například pomalého jednoduchého pohybu jako pomalé chůze a chaotický běh s přechodem do kotoulu a zběsilým máváním končetin. Předpokládám, že vzhledem k omezení Kinectů na 30 snímků za vteřinu budou mezi rychlým a pomalým pohybem velké rozdíly. Proto by nebylo směrodatné pro zjištění kvality snímání jednotlivých konfigurací proti sobě porovnávat rozdílné typy pohybu. Je důležité se pokusit pohyb udělat pro každou konfiguraci pokud možno co nejvíce stejně a také ho nasnímat za stejných podmínek. Nebylo by například vhodné porovnávat nasnímanou sekvenci, kdy byly pro Kinect zakryty veškeré lesklé povrchy a veškeré žluté plochy se špatnými informacemi o hloubce byly eliminovány proti sekvenci, která byla snímána v zrcadlovém bludišti.
2. **Nezasahovat do sekvencí ručně a porovnávat pouze surová získaná data;** Při zpracování sekvencí hodlám využít veškeré možnosti iPi Mocap studia, které má například funkci odstranění trasy z nasnímané sekvence, nebo možnost vylepšení již namapované sekvence na základního virtuálního

herce. Tyto akce iPi Mocap Studio propočítává a zpracovává automaticky, bez ručního zásahu do animace, a považuji je za nutnou část zpracování každé sekvence v iPi Mocap Studiu a po jejich použití považuji sekvence stále za v podstatě surový datový výstup, který zpracovával pouze automatický algoritmus. Do sekvencí ale nehodlám v této části práce vůbec zasahovat ručně, ať už v iPi Mocap Studiu nebo Blenderu, tím bych výsledky pro porovnání znehodnotil, protože by se již nejednalo o čistý záznam z Kinectů, ale pomáhal bych algoritmu zpracovávajícímu hloubková data z Kinectu ručním zásahem. IPi Mocap studio například umožňuje v případě, že se přestane v nějaký moment správně sledovat jedna končetina, tuto končetinu manuálně napozicovat správně a spustit sledování znovu z této nové pozice, tuto možnost už považuji za příliš velký manuální zásah a využívat jí nebudu.

3. **Všechny sekvence určené pro porovnávání použít na stejný 3D model;** Je nutné využít naprosto identický 3D model pro všechny porovnávací sekvence, protože pokud bych jednu sekvenci, která sledovala jednoho lidského herce, namapoval na model člověka a druhý na model chobotnice, rozhodně bych se nemohl dobrat jakýchkoli přijatelných závěrů při zhodnocení. Není možné využít ani dva rozdílné modely člověka, mohou mít rozdílné rigy s různým umístěním a počtem kostí a rozdílné skupiny různého počtu a pozice vertexů ve virtuálním 3D prostoru přiřazených rozdílnou mírou vlivu k jednotlivým kostem a při namapování dat ze získaných sekvencí by se mohly modely chovat a deformovat rozdílným způsobem. Je zkrátka nutné použít jeden jediný identický model pro všechny porovnávací sekvence.

5.2 Zvolené konfigurace pro porovnávání

Můj cíl je zhodnotit použití jednoho a více senzorů a Kinectů v1 proti Kinectům v2, proto jsem si zvolil pro porovnávání využít Mo-cap studio v následujících konfiguracích.

1. Jeden Kinect senzor v1 (Kinect for Windows)
2. Dva Kinect senzory v1 (Kinect for Windows)

3. Jeden Kinect senzor v2 (Xbox One Kinect)
4. Dva Kinect senzory v2 (Xbox One Kinect)

Tyto čtyři možnosti by měly být dostatečné pro získání relevantních výsledků na to, abych z nich mohl vyvodit nějaké závěry. S každou jednotlivou konfigurací nasnímám několik rozdílných typů pohybu, které proti sobě následně budu vyhodnocovat po zpracování iPi Mocap Studiem 4 a namapování na zvolený 3D model.

5.3 Kombinace využití Kinectů v1 a v2

Možnost využití Kinectů v1 a v2 není při snímání s iPi Recorderem bohužel možná a nepodařilo se mi najít software, který by takovou kombinaci umožňoval. Brekel nikde sice napsáno neměl, že by taková kombinace fungovat neměla, ale zprovoznit se mi ji nepodařilo, když jsem se o to pokusil.

Důvod proč toto zapojení není pravděpodobně možné, je, že Kinecty v1 a v2 mají rozdílné rozlišení obou kamer, a software s tím neumí pracovat, to je ale jen má domněnka. Každopádně při zapojení Kinectů v1 a v2 iPi Recorder vyžaduje zvolení pouze jednoho typu kamery a nedovolí nahrávání s oběma senzory v rámci jedné scény.

5.4 Vybrané typy pohybu pro porovávání

Před samotným snímáním jsem ještě musel vybrat, jaké pohyby vlastně hodlám snímat. Vybral jsem pět různých druhů pohybu, které mi přišlo vhodné vyzkoušet.

1. **Minimalistický pohyb** - Abych zjistil, jak dobře Kinect dokáže nasnímat herece, který je téměř statický, a zpracovat jen velmi drobné změny v pozici herece. V této sekvenci herec stojí na místě, jen mírně se pohybuje a předstírá, že postává v nějaké uličce, kde na někoho čeká a kouří cigaretu.
2. **Pomalý pohyb** - Abych zjistil, jak dobře Kinect snímá běžný pomalý pohyb. V této sekvenci herec pomalu přechází ze strany na stranu a poté udělá pomalý dřep.

3. **Rychlý pohyb** - Abych zjistil, jak dobře si Kinect poradí s rychlým pohybem, když dokáže snímat pouze 30 snímků za sekundu. V této sekvenci se herec pokouší o krátkou, velmi rychlou jump-stylovou taneční sestavu.
4. **Otáčení** - Abych zjistil, jak dobře Kinect dokáže snímat pohyb herce, který se kompletně celý otáčí, a postupně se senzoru zakrývají jeho končetiny, předpokládám, že zde budou největší rozdíly mezi použitím jednoho a dvou Kinect senzorů.
5. **Vzhůru nohama** - Může se to zdát zvláštní, ale na některých internetových fórech jsem si všiml, že někdo řešil problém, kdy byl Kinect zmatený, pokud se prohodila pozice rukou a nohou (když se dlaně ocitly pod úrovní chodidel). Zajímalo mě, zda je to pravda, a proto jsem sem tuto možnost přidal. V této sekvenci si herec lehne a své dlaně dá pod úroveň chodidel.

5.5 Příprava herce

IPi Mocap Studio využívá z Kinectů pro Mo-cap pouze data z hloubkových kamer, takže je jedno jakou barvu oblečení herec zvolí, je ale lepší zvolit přilnavé oblečení, které bude kopírovat postavu herce [30]. Osobně jsem zvolil těsné černé triko a legíny (viz. obrázek č.50), později jsem ještě přidal boty kvůli lepšímu snímání chodidel (tomu se budu věnovat dále v samostatné podkapitole). Pokud by herec měl například zimní bundu, Kinect by mohl mít problémy s rozpoznáním obrysů těla. Na oblečení by také neměly být žádné lesklé plochy, které by mohly Kinectu ztěžovat zjišťování dat o hloubce.



Obrázek 50: Herec v úplém oblečení kopírující křivky těla

Pokud při snímání budou využity motion controllery (nebo jiné senzory, které je nutné přidělat na tělo herce), je nutné je před začátkem snímání zkalibrovat a přidělat k tělu herce.[20]

V mém případě to byly tři Joy-cony, nemělo smysl využívat čtvrtý, protože iPi Mocap Studio 4 umí pomocí motion controllerů sledovat pouze dlaně a hlavu, teoreticky by šlo přidělat ke každé z těchto částí více senzorů a jejich výsledky poté zkombinovat, ale přišlo mi to zbytečné a nepraktické vzhledem k velikosti Joy-conů. Je nutné motion controllery zachytit k tělu tak, aby se sami o sobě nepohybovaly ani neotáčely a co nejpřesněji kopírovaly pohyb a rotaci zvolené části těla. Žádné oficiální příslušenství, které by je umožnilo připnout k tělu asi neexistuje, a tak je nutné využít vlastní tvořivost. Já jsem se rozhodl k dlaním Joy-cony přidělat s použitím gumy, která se dává do tepláků zakoupené ve švadlence a klasických gumiček (viz. obrázek č.51), aby se Joy-con nehýbal, potřeboval jsem okolo 5ti klasických gumiček a jednu gumu do tepláků. Gumičky se do dlaní poměrně ne-

příjemně po pár minutách zařezávali a proto jsem začal při snímání nosit tenké rukavice, což tento problém značně zredukovalo.



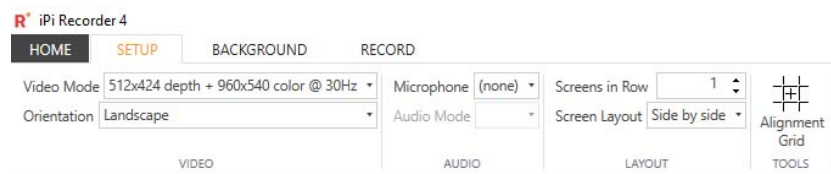
Obrázek 51: Uchycení Joy-conu k ruce

Připojení Joy-conu k hlavě bylo o něco těžší, Joy-con jsem chtěl připevnit na vršek hlavy a upevnit ho gumičkami přes krk, jako se zapíná například helma na kolo, to bylo ovšem dost nepříjemné a gumičky se v podstatě okamžitě bolestivě zařezávaly do krku. Zkusil jsem rozebrat čelovku a pásek z ní, který se navléká okolo obvodu hlavy, se ukázal jako vhodná možnost, pokud jsem si ho v prostoru,

kde tlačil do krku podložil složeným látkovým kapesníkem.

5.6 Nastavení nahrávání

Po výběru používaného hardwaru pro snímání se v iPi Recorderu otevře jeho hlavní část zaměřená na snímání na záložce setup, v ní se nachází pět kategorií nastavení (viz. obrázek č.52).



Obrázek 52: iPi Recorder základní nastavení

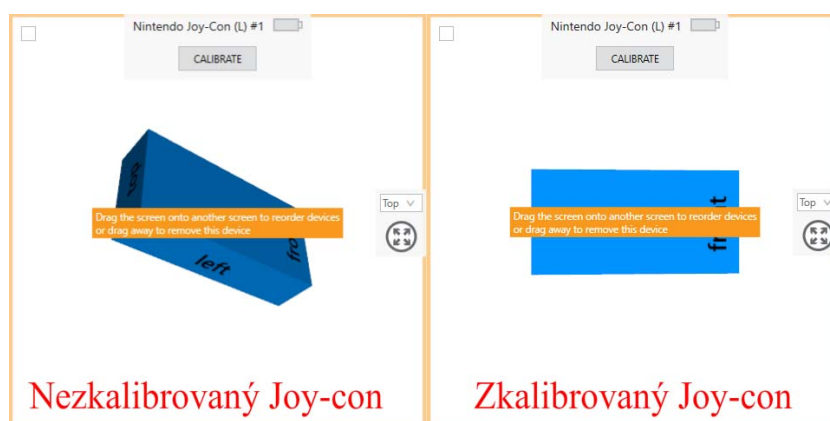
1. **Video** - zde se nachází nastavení video mode, které umožňuje změnit v jakém rozlišení a jaké kamery v Kinectu se budou při snímání používat. Protože data z RGB kamer nejsou potřeba, je možné je zde vypnout, hloubkové kamery se vypnout nedají, na Kinectech v1 a v2 mají jen jedno možné rozlišení. Já používal záznam z obou kamer, hlavně protože jsem ze záznamu z RGB kamer chtěl dělat screenshots do této práce. Druhá možnost v této kategorii je nastavení orientace obrazu, pokud je monitor z nějakého důvodu umístěn vertikálně nebo vzhůru nohama, je možné to tímto nastavením kompenzovat.
2. **Audio** - V této kategorii se nachází výběr mikrofону pro záznam zvuku a audio režim. Pro účely této práce je záznam zvuku irelevantní a není potřeba se mu věnovat.
3. **Shared settings** - Nachází se zde možnost vybrat hloubkový processor, snímání mi fungovalo nejlépe s výchozím nastavením, při výběru jiných možností se různě měnil dosažený framerate senzorů. O této možnosti nastavení se mi nepovedlo najít co přesně dělá, v podstatě nic se o ní nepíše ani na oficiálních stránkách programu ani na uživatelských fórech.
4. **Layout** - Zde je možné nastavit, jak se budou zobrazovat jednotlivé výstupy kamer a zařízení v iPi Recorderu. Možnost Screens in Row umožňuje nastavit,

kolik kamer se bude zobrazovat na jeden řádek (zvláštní je, že když Kinect snímá z obou svých kamer, iPi Recorder zde výstupy z obou bere jako jednu obrazovku), screen layout umožňuje vypnout zobrazení výstupu RGB kamer i pokud je Kinect používá, nebo RGB výstup umístit do pravého horního rohu obrazu hloubkové kamery a nebo je zobrazit vedle sebe, pokud se používají oba.

5. **Alignment grid** - Umožňuje zobrazit na každém výstupu kamery mřížku, která pomáhá zorientovat se v jednotlivých částech obrazu.

5.7 Kalibrace motion controllerů

Před každým nahrávacím sezením je Joy-cony potřeba zkalibrovat, jinak jsou data, které získají, nepoužitelná. Z mých zkušeností je potřeba kalibraci provádět každých cca 20 minut nahrávání, poté se začaly Joy-cony o několik stupňů odchylovat.



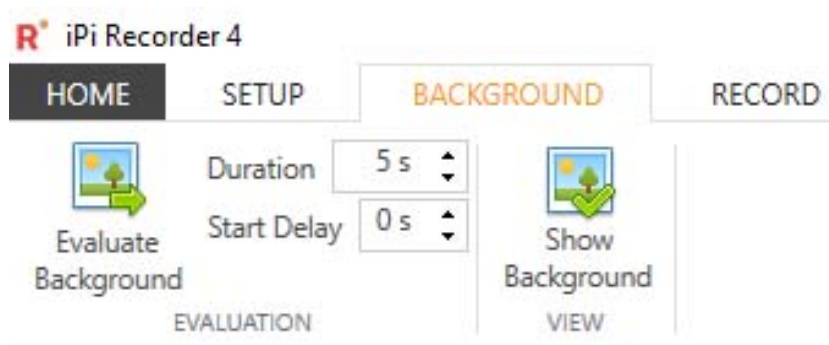
Obrázek 53: Kalibrace Joy-conů

Kalibrace Joy-conů je jednoduchá, po vybrání všech zařízení použitých pro danou scénu a stisknutí record video se dostanu na záložku setup, kde je možné kalibraci Joy-conů provést. Pravděpodobně se každý ovladač poprvé zobrazí v podivné pozici (viz. obrázek č.53), abych zjistil, který Joy-con je který, postupně vždy každý zvednu a zatočím s ním, podle toho, který začne na obrazovce rotovat, určím, o který se jedná. Joy-con se položí na rovný hladký povrch (například deska stolu) a v okně s daným Joy-conem je možnost calibrate, poté je nutné na

Joy-conu stisknout tlačítko pod analogovou páčkou (v případě levého Joy-conu se jedná o tlačítko screenshot, na pravém o tlačítko home), po jeho stisknutí proces kalibrace započne a měl by trvat několik vteřin. Joy-con by se měl nacházet ve zkalibrované vodorovné pozici (viz. obrázek č.53). Tento postup je nutné opakovat pro každý Joy-con.

5.8 Evaluace pozadí scény

Další záložkou po setup v iPi Recorderu je background (viz. obrázek č.54). Před každým novým snímáním, pokud se s Kinecty pohybovalo nebo se změnilo cokoli v zorném poli Kinectů, je nutné provést evaluaci pozadí každého Kinectu.

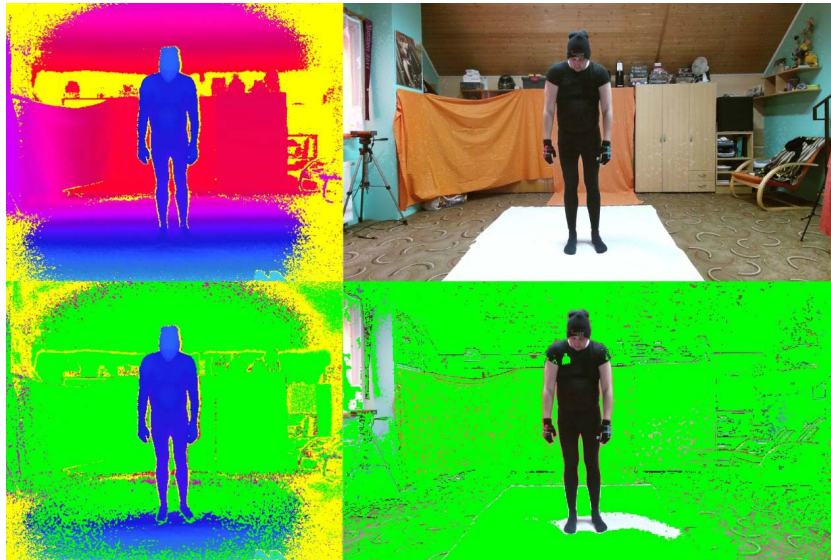


Obrázek 54: iPi Recorder 4 - Evaluace pozadí

Získané pozadí je souborem všech statických objektů v zorném poli Kinectů. Evaluované pozadí je vloženo uvnitř nasnímaných videí a později v iPi Mocap studiu pomáhá při rozpoznávání pohybujících se objektů. V oblasti se nesmí vyskytovat žádné pohybující se objekty jako lidé, nebo objekty, které tam potom nebudou a bránily by výhledu na část pozadí, která se za objektem nachází. V takovém případě by nebylo možné správně odlišit pozadí od herce. [31]

V kategorii evaluation jsou možnosti nastavit délku evaluace pozadí (Duration), toto nastavení udává po jakou dobu se Kinect bude snažit pozadí pozorovat a ovlivňuje kvalitu výsledného získaného pozadí (pro většinu situací postačí pět sekund), možnost start delay umožňuje nastavit zpoždění, po kterém se začne pozadí snímat, v případě, že je například řídící počítač s člověkem v záběru, a člověk za počítačem se musí dostat ze záběru před spuštěním evaluace. V kategorii view je

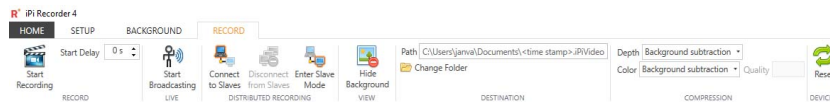
možnost zobrazit evaluované pozadí.



Obrázek 55: Aplikace evaluovaného pozadí při klíčování

5.9 Nastavení nahrávání

Po evaluaci pozadí je v iPi Recorderu další záložkou nahrávání (Record), v ní se nachází následující kategorie.



Obrázek 56: iPi Recorder 4 - nastavení nahrávání

1. **Record** - Zde je možné nastavit zpoždění spuštění nahrávání a spustit nahrávání.
2. **Live** - Zde je možné spustit vysílání pro snímání v reálném čase v iPi Mocap Studiu.
3. **Distributed Recording** - Umožňuje nastavení Master-Slaves snímání, to už jsem podrobně probíral dříve v kapitole, která se přímo věnovala Master-Slave módu a jeho porovnání se zapojením všech Kinectů do jednoho počítače.

4. **View** - Umožní skrytí pozadí pomocí evaluovaného pozadí, pokud byla evaluace dostatečně kvalitní, měl by zůstat ve scéně pouze samotný herec.
 5. **Destination** - Zde uživatel může zvolit, kam ukládat nasnímané sekvence.
 6. **Compression** - Nastavení komprese pro RGB a hloubkové kamery. Není třeba ho měnit, nabízí se pouze možnost žádné komprese nebo odstranění pozadí.
7. **Devices** - Umožní resetování všech zařízení používaných při snímání.

5.10 Kalibrace scény

Provést kalibraci scény je nutné pouze, pokud budu snímat s více než jedním Kinectem. Je dobré při rozestavení senzorů dodržet několik zásad, na které jsem během svých pokusů o Mo-cap přišel. Pokud je v plánu použít dva senzory v úhlu okolo 180 ° a provádět kalibraci pomocí 3D plochy, pomůže mít senzory v naprosto stejné výšce co nejvíc naproti sobě a v přesné rovině bez jakéhokoliv náklonu. Tím se značně může zjednodušit proces manuálního pozicování při kalibraci v iPi Mocap studiu, které v základu předpokládá, že senzory budou v tomto rozestavení.

Toho jsem dosáhl použitím stativů a metrem si ověřil, že jsou ve stejné výšce, s použitím vodováhy se ujistil, že senzory jsou v rovině, a případně vypodložil nohu stativu, pokud nebyly. Naklonění Kinectů se u Kinectů v1 dá ovládat přímo z iPi Mocap Studia, takže stačí nastavit výchozí hodnotu, u Kinectů v2 to musím provést manuálně a zkusit náklon trefit okem.

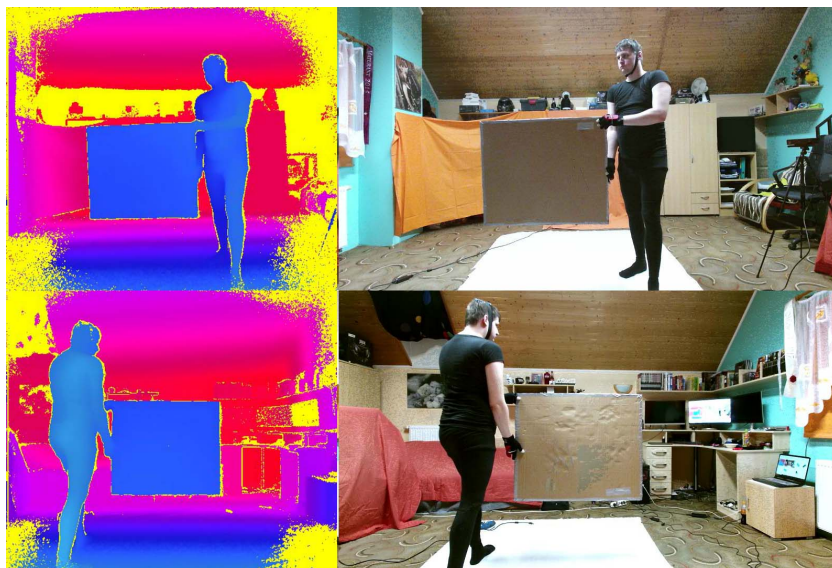
V iPi Mocap Studiu existují dvě možnosti kalibrace, jedna starší pomocí 3D plochy a druhá pomocí světelné značky.

5.10.1 Kalibrace pomocí 3D plochy

3D plocha; Tento druh kalibrace je starší a je funkční pouze pro kalibraci dvou Kinect senzorů. Potřebuji k němu nějakou nepříliš tlustou obdélníkovou nebo čtvercovou plochu o velikosti cca 0,7 x 0,7 m až 1,5 x 1,5 m o tloušťce okolo max 1 cm. Může se jednat například o nějakou dřevěnou desku, nebo kus kartonu.

Já se rozhodl vyrobit si kalibrační desku ze staré krabice od kola, ze které jsem vyřízl dva stejně velké kusy, a přidělal je k sobě, aby se karton při manipulaci tolik neprohýbal.

Nahrání kalibrační sekvence; Po spuštění nahrávání v iPi Recorderu se herec postaví s kalibrační deskou mezi dva Kinect senzory tak, aby byla deska kompletně viditelná pro hloubkové senzory obou Kinectů (viz. obrázek č.57). Následně se začne pohybovat snímacím prostorem s deskou stále viditelnou pro oba senzory. [21] Délka kalibrační sekvence, kde je deska viditelná, by měla být přinejmenším pět sekund, čím delší bude, tím je větší pravděpodobnost, že se v sekvenci najde dobrý úsek vhodný pro kalibraci.



Obrázek 57: Snímání kalibrační sekvence pomocí 3D plochy

Je důležité, aby kalibrační sekvence měla stejné rozlišení a snímky za vteřinu (ty mohou být vyšší), jako sekvence obsahující pohybujícího se herce. Nižší snímky za vteřinu by způsobily problémy se synchronizací.

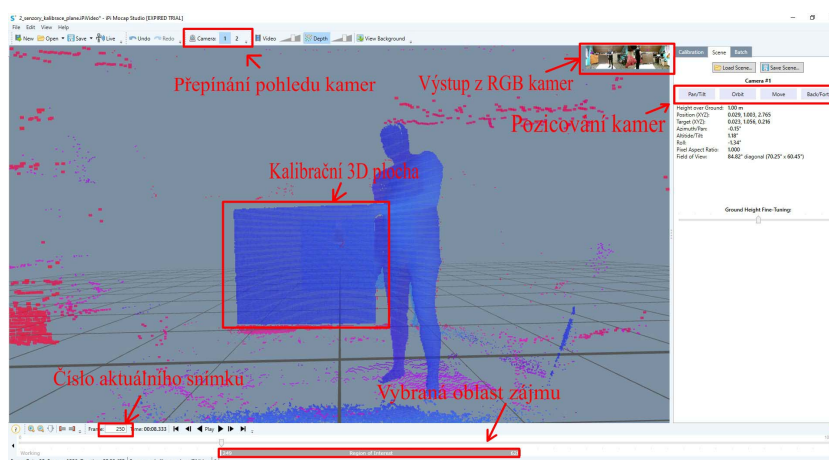
Proces kalibrace pomocí 3D plochy; Při otevírání nasnímané sekvence obsahující data z více Kinect senzorů se iPi Mocap Studio zeptá, zda se bude jednat o kalibrační projekt nebo projekt s nahranou akcí, v tomto případě chci kalibrační.

Tím se dostanu do hlavního prostředí iPi Mocap Studio, jako první chci izolovat část sekvence, kterou budu chtít použít pro kalibraci. K tomu slouží spodní část okna programu, část s nápisem region of interest označuje vybranou část sekvence, zmenším ji tedy tak, aby zahrnovala pouze část určenou ke kalibraci.

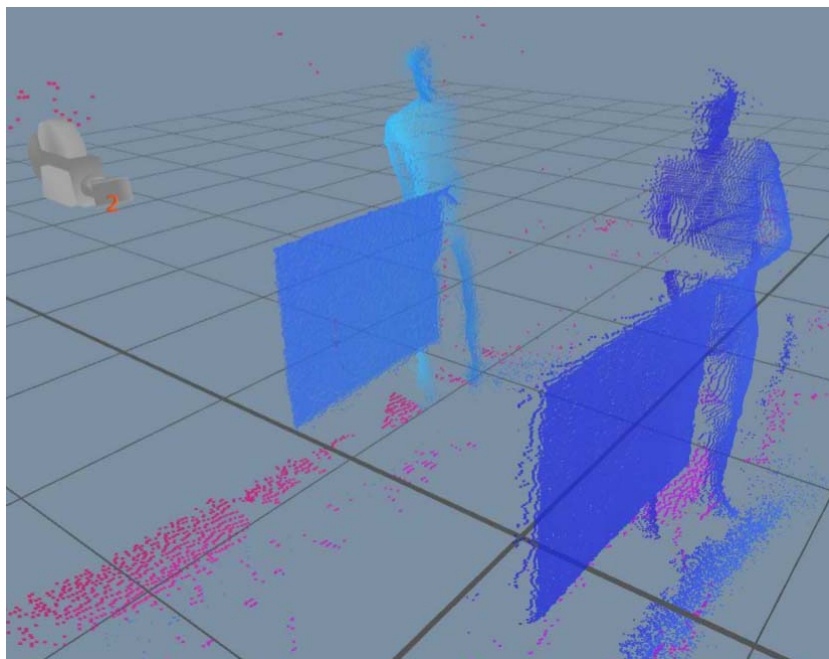
Při kalibraci s 3D plochou budu potřebovat vidět data ze všech senzorů najednou, ale iPi Mocap studio defaultně zobrazuje pouze data z aktuálně vybraného senzoru. To se dá změnit v záložce view - show depth from all sensors.

iPi Mocap Studio předpokládá rozložení senzorů přímo naproti sobě, pokud tak byly skutečně rozloženy, hloubková data se zobrazí z obou poměrně přesně a bude snazší je sladit, pokud byly senzory v úplně jiném rozpoložení, může scéna vypadat poměrně chaoticky a její srovnání bude nejspíš složitější.

V horní části okna programu pod kategorií Camera lze přepínat mezi pohledem z jednotlivých senzorů, což je pro tento typ kalibrace klíčové při srovnávání hloubkových dat. V kategorii video je možné vypnout a zapnout zobrazení výstupu RGB kamer a nastavit jeho průhlednost a v kategorii Depth to samé s výstupem hloubkových kamer. Mě zajímají pouze hloubková data, a proto klasické video vypnu.

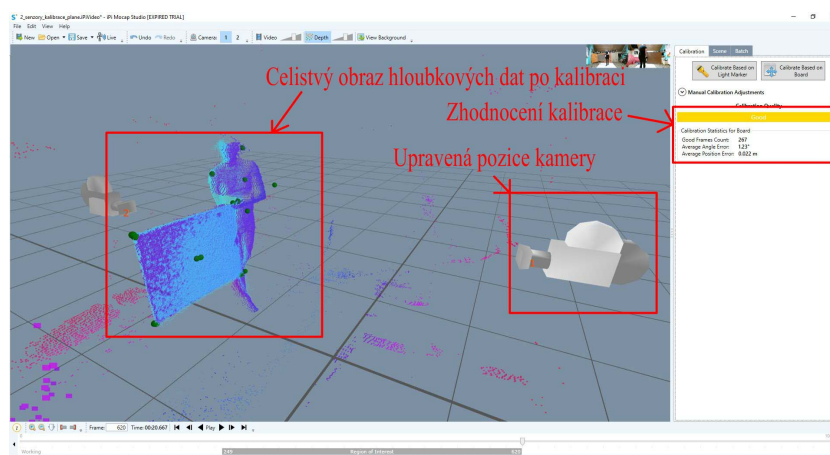


Obrázek 58: iPi Mocap Studio - okno programu při kalibraci s 3D plochou



Obrázek 59: Hloubková data z dvou senzorů před kalibrací

V pravé části okna programu se nachází záložka scene, zde je možné rotovat, pohybovat a naklánět aktuálně zobrazovaný výstup senzoru. Mým cílem je pomocí těchto možností co nejpřesněji sladit hloubková data z obou senzorů pro jeden ucelený obraz. Po jeho dosažení si sekvenci přehraji a pokud se po celou dobu obraz zdá celistvý a nerozpadá se, přejdu do záložky Calibration a vyberu možnost Calibrate Based on Board. Program se nyní pokusí propočítat přesné umístění kamer, které jsem použil při nahrávání sekvencí, to bude trvat podle toho, jak dlouhá je vybraná kalibrační oblast a jak výkonný je samotný počítač.



Obrázek 60: Úspěšná kalibrace s pomocí 3D plochy

Po skončení procesu kalibrace program v pravém panelu vypíše, zda byla kalibrace úspěšná (jsou tři možnosti - Failed, Good a Perfect) s informacemi o počtu snímků, ve kterých byl schopný sledovat potřebné referenční body, a byly správně sladěné pro oba senzory (v každém snímku kde jsou správně, se zobrazují jako zelené koule na 3D ploše, mohou být i žluté a červené, pokud je nelze sladit nebo jsou vyloženy špatně), informace o průměrné odchylce pozice a úhlu (viz. obrázek č.60).

Po skončení kalibrace by měly být značky představující senzory ve virtuálním prostoru umístěné přesně tak, jako byly při reálném snímání.

Podle přesnosti kalibrace se poté zobrazují data při zpracování scén s akcí, a i když je možné využít kalibraci s menší přesností, kterou program označuje jako Good, osobně bych ji nepoužil, a vždy se snažil dosáhnout pouze perfektní kalibrace se zcela minimálními odchylkami.

Pokud kalibrace nebyla úspěšná, je možné se pokusit upravit znovu pozice dat z hloubkových sensorů a provést ji znovu nebo vybrat i jiný úsek sekvence určeného pro kalibraci, pokud je sekvence dostatečně dlouhá.

Když je kalibrace hotová, zbývá jen scénu uložit, vytvoří se soubor s koncovkou iPiCalib, který se použije při zpracování akčních sekvencí.

5.10.2 Kalibrace pomocí světelné značky

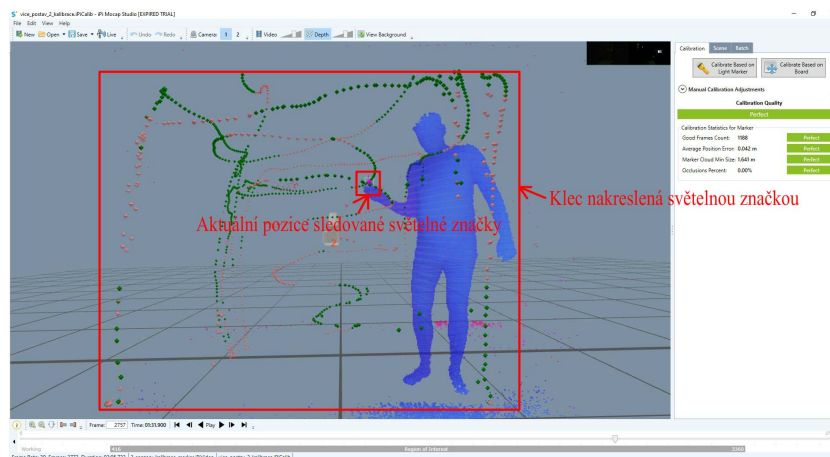
Kalibrace pomocí světelné značky je novější možnost přidaná do iPi Mocap Studia a umožňuje kalibraci dvou a více senzorů. Při této variantě je třeba brát ohled na světlo, neboť senzory budou sledovat světelnou značku, kterou herec drží v ruce, a pokud je silně intenzivních světelných bodů v záběru víc, senzor tím může být zmatený a přijímat špatné informace.



Obrázek 61: Upravená svítilna pro použití jako světelná značka

Postup je až k započetí nahrávání prakticky identický, herec po spuštění nahrá-

vání přijde na scénu se svítící značkou v ruce, může to být například starší baterka s odšroubovaným vrchem, světlo nesmí svítit jen jedním směrem, ale do všech stran. Herec se pohybuje po snímaném prostoru a kreslí v něm pomocí světelného markeru něco jako klec, měl by se pokrýt celý hlavní snímaný prostor. Je nutné, aby světelnou značku viděly všechny senzory, jinak nebude kalibrace možná.[32]



Obrázek 62: Kalibrace pomocí světelné značky

Při vytvoření kalibračního projektu v iPi Mocap Studiu jsem opět zredukoval point of interest na část sekvence, kterou chci využít ke kalibraci, je možné jako v předchozí metodě zhruba upravit pozici senzorů, aby rovnou co nejlépe refletovaly stejné rozložení jako při reálném snímání, ale není to většinou nutné. Jako metodu kalibrace tentokrát zvolím Calibrate Based on Light Marker a software se pokusí v každém snímku sledovat světelný bod. Je doporučeno použít sekvenci o délce několik stovek snímků, ale nikoli v řádech tisíců, takže při 30 fps může být sekvence s délkou okolo 30 sekund ideální. Po skončení kalibrace software opět vypíše, jestli byla úspěšná, a jaké byly odchylky, v případě neúspěchu je možné pokusit se senzory sladit manuálně a spustit kalibraci znovu.

5.10.3 Zhodnocení kalibrace

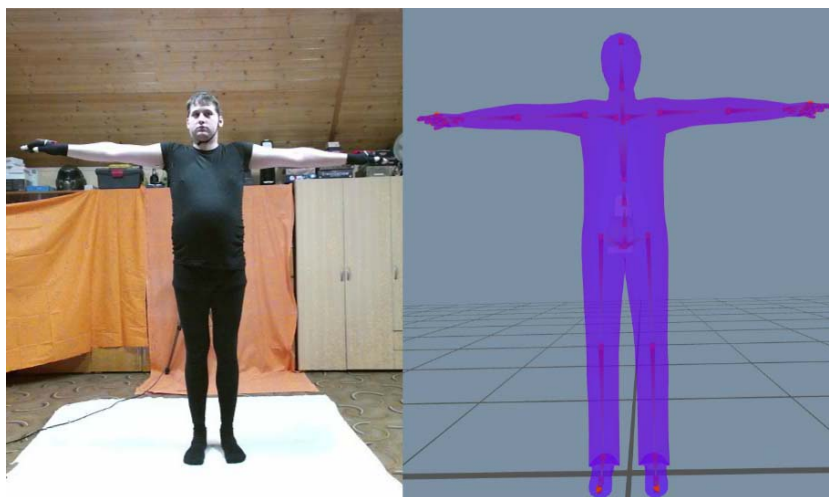
Osobně nemám kalibraci pomocí 3D plochy v oblibě, je zdoluhavá, manuální pozicování senzorů je otravné a složité a dosáhnout perfektní kalibrace je někdy téměř nemožné. Několikrát jsem se pokusil při kalibraci s 3D plochou marně o kalibraci,

až jsem kalibrační sekvenci musel nahrávat znovu, což je velmi otravné a zdržuje práci a stejně jsem sotva dosáhl přijatelné kalibrace. Tato metoda navíc funguje pouze na kalibraci dvou senzorů a není možné s ní zkalibrovat jinou konfiguraci.

Kalibrace se světelnou značkou je z velké části automatická, jen je potřeba pohlídat si světelné podmínky v místnosti a mít něco, co lze použít jako světelnou značku. Kalibrace byla většinou vyhodnocena bez jakékoli úpravy jako perfektní s velmi malou odchylkou a tato metoda funguje dobře i pro více než dva Kinecty.

5.11 T-Pose

Začátek snímání by měl herec vždy začínat v T-pose [21], jedná se o zažitý standart a téměř každý standartní 3D humanoidní model jí má jako výchozí pózu, i v iPi studiu je jeho základní Rig v T-pose, takže je vhodné na ní na začátku namapovat nasnímaná data.



Obrázek 63: Skutečný a virtuální herec v T-pose

Herec se postaví čelem k hlavnímu Kinect senzoru, dá nohy k sobě, postaví se rovně s pohledem přímo na senzor a ruce rozpřáhne podél těla v úhlu 90° stupňů, jako je na obrázku č.63. V této póze je potřeba setrvat pouze vteřinu a poté herec může předvádět pohyb, jaký chce nasnímat.

Kromě T-Pose existuje ještě A-Pose, kdy ruce postavy místo úhlu 90° jsou v úhlu okolo 45° , občas se dá narazit na modely, které jsou v A-Pose a může být

potřeba je do T-Pose pro tyto účely upravit.

5.12 Snímání sekvencí pro účely porovnávání

Rozhodl jsem se jednotlivé akce nasnímat samostatně, místo, abych je nasnímal jako jednu sekvenci s několika akcemi, oba způsoby jsou validní, já osobně mám radši všechno rozdělené spíše na menší části, se kterými můžu pracovat samostatně. Také jsem se rozhodl vždy nejdříve udělat celou kalibraci místo nahrání kalibrační sekvence a akčních scén naráz. Tím jsem eliminoval možnost, že by kalibrace nebyla možná provést perfektně z nahrané sekvence, a musel bych nahrávat i všechny scény s akcí pro danou konfiguraci znovu.

5.12.1 Snímání s jedním Kinect senzorem

Protože Kinect v1 nemá úchyt k připojení na stativ, namontoval jsem na něj Kinect v2 a na něj položil Kinect v1 (viz. obrázek č.64), tím pádem jsem při snímání z obou senzorů měl téměř identickou část scény a nemusel se bát, že by mi výsledky ovlivňovalo snímání z jiného úhlu a pozice každým senzorem odděleně a také to vyřešilo problém s tím, jak rozmístit Kinecty v1. Při nahrávání těchto sekvencí jsem měl vše zapojené do jednoho PC, jak jsem psal již dříve v práci, nejsem velkým fanouškem Master-slave pokud není nutné ho využít.



Obrázek 64: Kinect senzory v2 a v1 na stativu

Při snímání jedním senzorem se nedělá kalibrace senzorů, takže po kalibraci Joy-conů a poté, co jsem se jako herec připravil a umístil na sebe motion controllery, jsem spustil zpožděné nahrávání v iPi Recorderu o deset sekund, což mi stačilo na to vstát od počítače a dojít do oblasti snímání a zaujmout T-Pose. Při využití motion controllerů je nutné, aby každý z nich alespoň v jednom snímku sekvence směřoval přímo na Kinect senzor [20], a proto jsem vždy po T-Pose postupně hned namířil všemi senzory přímo na Kinect, s motion controllerem na hlavě je to trochu těžké odhadnout, ale aspoň přibližně je možné se trefit.

Poté co se nahrávání spustilo, jsem nahrál vždy nejdřív sekvenci s minimalistickým pohybem, který jsem popsal dříve společně s ostatními druhy snímaného pohybu, poté jsem nahrávání vypnul, získanou sekvenci si pojmenoval tak, abych vždy věděl, co se v ní přesně nachází, a s jakou konfigurací byla natočena a hned nastavil další nahrávání se zpožděním a proces opakoval pro všech pět typů pohybu.

Po nahrání všech potřebných sekvencí s Kinectem v1 jsem v iPi Recorderu přenastavil snímání na Kinect v2, provedl novou evaluaci pozadí a opakoval všechny přechozí kroky, dokud jsem neměl všech pět sekvencí i z tohoto senzoru.

Všiml jsem si, že snímací plocha je díky většímu zornému poli Kinectů v2 značně rozšířená, a herec se může pohybovat více do stran než u Kinectu v1.

5.12.2 Snímání se dvěma Kinect senzory

Když jsem měl nasnímané všechny potřebné sekvence pro konfigurace s jedním senzorem, umístil jsem druhý stativ s Kinecty v2 a v1 do vhodné pozice naproti prvnímu stativu s Kinecty a zapojil senzory do počítače.

V iPi Recorderu jsem přenastavil vybrané snímací zařízení na oba Kinecty v1 a tři Joy-cony, které jsem měl přidělané k tělu. Všiml jsem si, že dva Joy-cony už začínají driftovat, a je potřeba provést znovu jejich kalibraci, takže jsem je musel sundat a znovu zkalibrovat a opět si je upevnit na dlaně. Opět jsem provedl evaluaci pozadí pro oba senzory a spustil zpožděné nahrávání. To jsem musel hned přerušit, protože nebyla možnost si stoupnout tak, aby byla celá postava herce v záběru obou hloubkových sensorů. Rozhodl jsem se nakonec oba senzory naklonit tak, aby se jeden zaměřoval více na vrchní polovinu těla a druhý na sledování nohou, náklon to byl lehký pouze o několik stupňů, ale poskytl mi dostatek prostoru pro manévrování po snímací ploše, a aby všechny části těla zůstaly v záběru alespoň jednoho ze sensorů, z jednoho mohly chybět části nohou a z druhého zase hlava a krk, ale torzo a ruce byly v podstatě neustále v záběru obou při pohybu po celé scéně.

Po této úpravě jsem mohl konečně nahrát všechny potřebné sekvence s touto konfigurací. Při snímání dvěma senzory jsem měl k dispozici menší prostor na pohyb po ose mezi senzory, protože jejich slepé pole okolo 40ti centimetrů od senzoru by mě nesnímalo, takže jsem se nemohl k ani jednomu do této vzdálenosti přibližovat. Na rozdíl od snímání s jedním senzorem jsem po této ose měl tedy cca o 60 cm menší oblast (40 centimetrů slepé pole Kinect + plocha zabraná stativem). Protože jsem nahrával s více senzory, bylo nejdříve potřeba nahrát kalibrační sekvenci, při které jsem se rozhodl použít metodu světelné značky. Kalibraci jsem rovnou zpracoval, a když program vypsalo, že je perfektní, rovnou jsem si ji uložil pro pozdější použití při zpracování sekvencí s akcí. Poté jsem opět spustil nahrávání a nahrál všechny potřebné akční sekvence. Po nahrání všech potřebných sekvencí jsem vypojil z počítače Kinecty v1 a zapojil oba Kinecty v2 a proces opakovat.

Při snímání se dvěma Kinecty v2 jsem díky většímu zornému poli obou senzorů mohl vměstnat celou postavu do záběru v obou Kinectech a nemusel téměř vůbec upravovat jejich náklon. Rozhodl jsem se ho upravit jen zcela minimálně směrem dolů, aby zabíraly větší část podlahy pro lepší sledování pohybu nohou, protože zabíraly zbytečně moc velkou část stropu, kam se moje postava jako herce stejně při snímání těchto sekvencí nemohla dostat. Opět jsem musel nahrát kalibrační sekvenci se světelnou značkou a zpracovat ji a poté nahrát sekvence s akcí. Při nahrávání sekvencí se dvěma Kinecty v2 se mi občas objevil problém s nižšími snímky za vteřinu, vždy ho vyřešilo přesunutí Kinectu do jiného USB 3.0 portu, jinak se snímání obešlo bez dalších problémů.

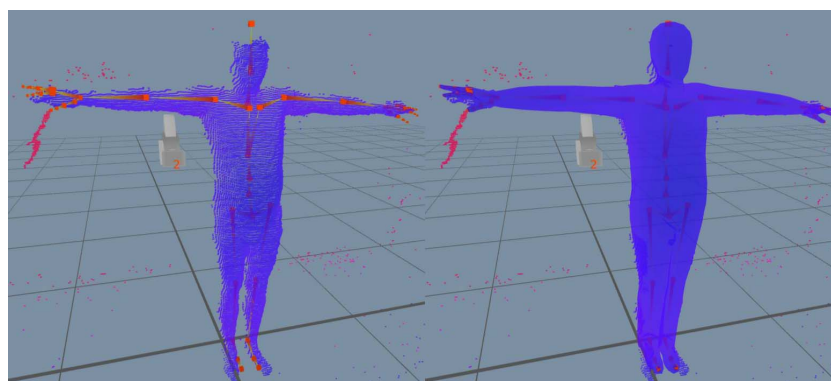
Po skončení snímání jsem měl nasnímaných celkem 22 sekvencí, pět s akcí pro každou konfiguraci a dvě kalibrační pro obě sestavy se dvěma senzory. Nyní jsem se konečně mohl pustit do samotného zpracování sekvencí s akcí iPi Mocap Studiem.

5.13 Zpracování sekvencí v iPi Mocap Studiu

5.13.1 Sekvence s jedním Kinectem

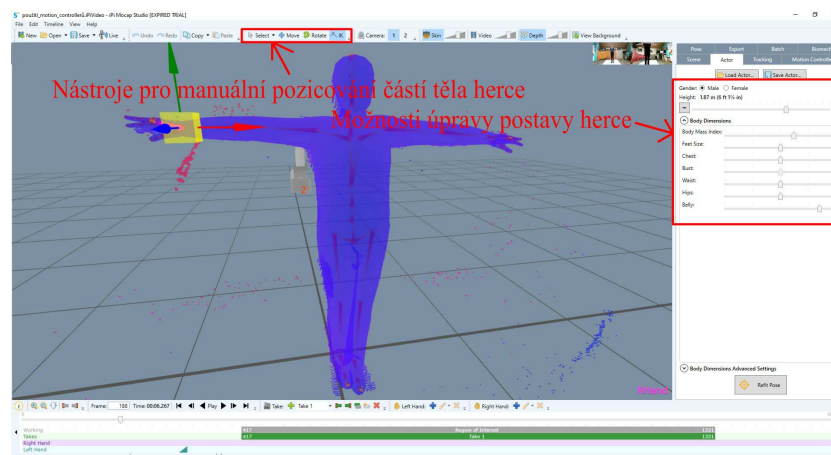
Při vytvoření nového projektu v iPi Mocap Studiu jsem vybral sekvenci obsahující akci, kterou jsem chtěl zpracovat, a poté vybrat přibližnou výšku snímaného herce, čemuž iPi Mocap Studio uzpůsobí vlastní model virtuálního herce. Stejně jako při zpracování kalibrační sekvence jsem ve spodní části okna programu musel upravit, s jakou částí sekvence vlastně chci pracovat.

Zde je rozdíl, že místo jednoho posuvníku jsou zde dva, spodní označená zeleně je Take 1 a tím označím, jakou část videa budu celkově zpracovávat, tento posuvník v průběhu zpracování sekvence již dále měnit nebudu. Vždy jsem jeho začátek posunul tam, kde postava herce byla v co nejpřesnější T-pose a jeho konec ihned po dokončení pohybu, který jsem v dané sekvenci chtěl zpracovávat (vždy bylo potřeba odříznout část, kdy jsem šel vypnout snímání k počítači). Druhý posuvník je Region of Interest a ten označuje pro jakou část vybrané sekvence provádím jednotlivé úpravy a tracking, při prvotním zpracování a trackingu celého pohybu ho nechám ve stejné pozici jako posuvník pro Take 1, ovšem pokud je potřeba například dále provést jen částečný tracking některé části těla pro opravu původního trackingu, upravím Region of Interest pouze na požadovanou část. [30]



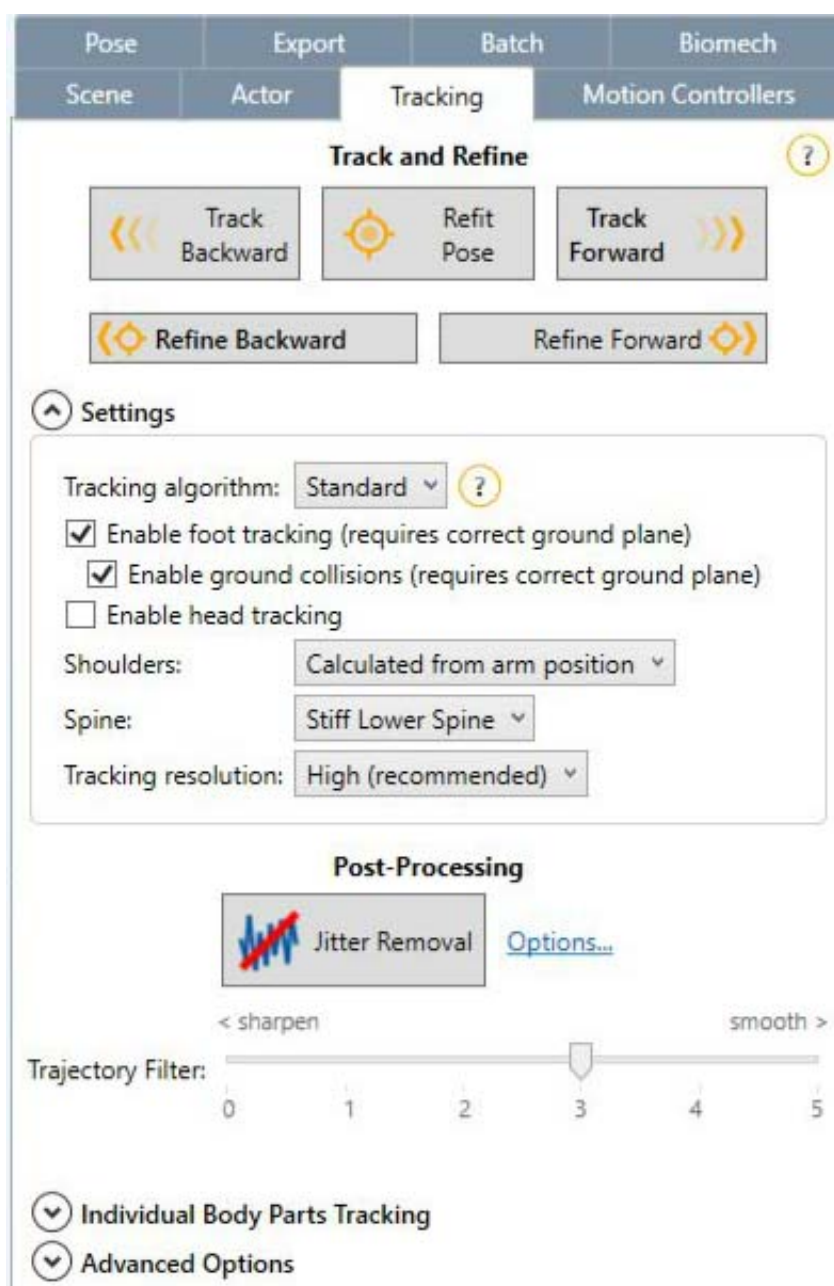
Obrázek 65: Zarovnání virtuálního herce s hloubkovými daty

Stejně jako při kalibraci jsem si vždy vypnul zobrazení video výstupu a ponechal pouze hloubková data ze všech senzorů a také Rig herce s jeho průhledným modelem (je možno nastavit v kategorii skin). Poté je nutné hloubková data představující herce sladit s modelem herce iPi Mocap Studia (viz. obrázek č.65). Toho je možné dosáhnout pomocí nástrojů Move, Rotate a IK (Inverse Kinematics), kdy se pohybuje s celým modelem herce a jeho jednotlivými částmi ovládanými rigem, dokud není co nejpřesněji sladěný s nasnímanou postavou herce. V záložce Actor v pravé části programu je možné model virtuálního herce přizpůsobit, aby co nejlépe odpovídal postavě reálného herce (viz obrázek č.66), lze změnit jeho pohlaví, výšku, BMI, velikost chodidel, hrudí, pas a břicho. Pozici všech částí modelu jsem zkoumal ze všech úhlů ve virtuálním 3D prostoru a snažil se je co nejpřesněji zarovnat po všech osách.



Obrázek 66: iPi Mocap Studio - nástroje pro úpravu pozice a postavy virtuálního herce

Přesunul jsem se na pravém panelu do záložky Scene, kde jsem pomocí posuvníku Ground Height Fine-Tuning upravil pozici podlahy, pokud bylo potřeba aby odpovídala pozici chodidel (což většinou nebylo potřeba, protože jsem měl scénu správně nastavenou a podlahu rozpoznali správně při nahrávání). Také jsem zde vycentroval scénu na charakter v prvním vybraném snímku zpracovávané sekvence a pokud by s tím automatika programu měla problém, je možné vystředění provést i manuálně pomocí Move Coordinate System.



Obrázek 67: iPi Mocap Studio - nastavení trackingu

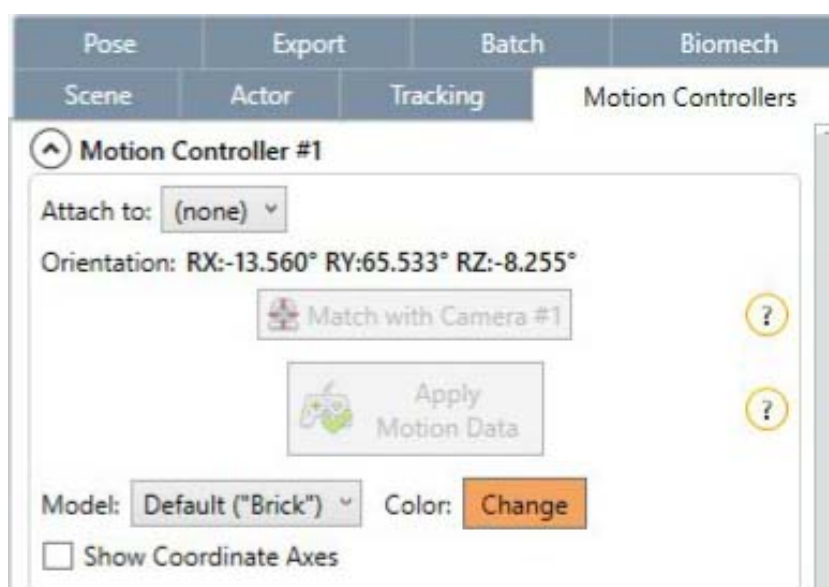
Následně jsem se přesunul do záložky Tracking (viz. obrázek č.67)kde se nachází následující možnosti:

1. **Track Backward a Track Forward** - Spustí tracking pohybu od aktuálního

vybraného snímku z aktuální pózy virtuálního herce směrem vpřed nebo vzad ve vybrané oblasti zájmu. Jedná se o primární tracking, který se následně vylepšuje. Je důležité, aby byl před spuštěním v aktuálním snímku model virtuálního herce co nejpřesněji zarovnaný s hloubkovými daty.

2. **Refit pose** - Při úpravách z prvotního trackingu, pokud je póza manuálně upravovaná, se pokusí jí co nejlépe zarovnat s hloubkovými daty a následujícími/předchozími snímky.
3. **Refine Backward a Refine Forward** - Spustí nový tracking, který má vylepšit původní primární tracking směrem vpřed nebo vzad od aktuálního snímku ve vybrané oblasti zájmu. Dokáže automaticky opravit menší chyby v primárním trackingu a je o něco pomalejší než primární tracking. Měl by být aplikovaný pouze se stejnými parametry jako primární tracking a měl by být aplikovaný před použitím dat z motion controllerů. Na rozdíl od primárního trackingu se nesnaží dělat žádnou predikci pózy a provádí své výpočty vždy pouze pro aktuální snímek. [30]
4. **Settings** - Zde je možné nastavit jaký algoritmus se pro tracking použije, standart je pro finální tracking a rychlý pro účely testování. Je zde možné vypnout a zapnout tracking chodidel, kolize herce s podlahou, pohybu hlavy, způsob trackingu ramen, ztuhlost nebo flexibilitu spodní části páteře a rozlišení trackingu.
5. **Post-processing** - Jitter Removal se stará o vyfiltrování nechtěného šumu ve vybrané oblasti zájmu a na vybraných částech modelu herce, který způsobuje třas při pohybu modelu. [30] Je potřeba ho využít prakticky vždycky před exportem animace. V jeho nastavení můžu upravit jeho sílu pro každou jednotlivou část těla virtuálního herce. Trajectory filter se stará o odstranění zbývajícího jemného šumu po aplikaci Jitter Removal, jeho aplikace je velmi rychlá a sám se aplikuje vždy přímo za běhu a je doporučené nechat ho na výchozí hodnotě.
6. **Individual Body Parts Tracking** - Je zde možné vybrat, s kterými částmi těla se bude momentálně pracovat.
7. **Advanced Options** - Umožňuje aplikovat starou verzi Jitter Removal.

Vždy jsem jako první spustil track forward od prvního snímku zpracovávaného pohybu a poté si výsledek několikrát přehrál a hledal chyby v prvotním trackingu. Nechal jsem zapnutý tracking chodidel a kolízí s podlahou a vypnutý head tracking, protože na něj jsem používal motion controller, pozici ramen jsem nechal propočítávat z pozice rukou, spodní páteř jsem nastavil na flexibilní a rozlišení jsem vždy nechal vysoké. V rámci zjišťování kvality snímání jednotlivých konfigurací Kinectů jsem neprováděl žádné manuální zásahy do chyb v trackingu (například, když se přestal sledovat pohyb některé končetiny a podobně), jinak bych normálně v této chvíli v částech, kde se vyskytl problém s trackingem, manuálně upravil pozici špatně trackované části těla v posledním snímku, kde se chyba vyskytuje, dal Refit Pose a provedl tracking dané části těla směrem dozadu do části, kde se chyba vyskytla prvně (pomocí posuvníku Region of Interest). Poté jsem spustil Refine Forward od prvního snímku pro vylepšení prvotního trackingu.



Obrázek 68: iPi Mocap Studio - nastavení motion controllerů

Poté jsem se přesunul do záložky Motion Controllers (viz. obrázek č.68), kde jsem potřeboval přiřadit použité Motion Controllery k jednotlivým částem těla a aplikovat na ně jejich pohybová data. Každý motion controller je defaultně zobrazován jako barevná kostka rotující pod podlahou. Která kostka patří ke kterému ovladači se pozná podle její barvy, kterou je možné nastavit v okně každého motion

controlleru a také podle samotného modelu, který lze změnit na meč, pistol, PS Move nebo jeho model zcela skrýt. Který motion controller patřil ke které dlani nebo hlavě jsem poznal podle chování rotace každé kostky, v sekvenci jsem vždy nechal část, kdy jsem jednotlivými Joy-cony zamířil přímo na Kinect a zakroutil dlaní do stran, takže kostka, která v tu chvíli rotovala, patřila k dané dlani, tímto způsobem jsem postupně všechny kostky připojil k oběma dlaním a hlavě a jejich modely se poté přesunuly na dané části těla virtuálního herce.

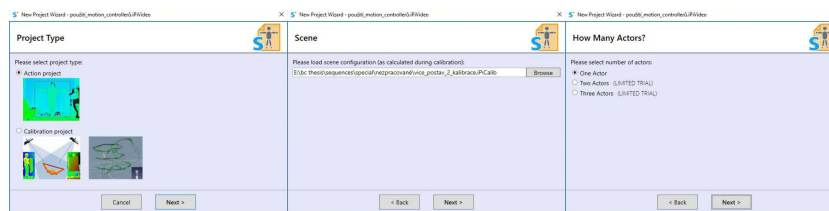
Potom se musí sladit jejich pozice s kamerou, aby se daly správně aplikovat jejich data pro pohyb dlaní a rotaci hlavy. Kvůli tomu jsem při nahrávání mířil vždy v jedné chvíli každým motion controllerem přímo na Kinect (já vždy mířil na primární Kinect i při snímání s více Kinecty, který byl označený jako senzor #1, ale pokud bych omylem mířil na druhý, je možné se přepnout do jeho pohledu a motion senzory sladit s ním), našel jsem daný moment, kdy senzor mířil v nahrané sekvenci přímo na senzor a pro daný motion controller stiskl Match with camera #1. Tím se kostka představující motion controller v tu chvíli srovnala směrem ke kameře.

Poté jsem aplikoval jejich získaná data pomocí Apply Motion Data a tím získal mnohem přesnější rotaci dlaní a hlavy oproti rotaci, kterou dokázaly zpracovat samotné Kinecty. V několika sekvencích se stalo, že kalibrace motion controllerů nebyla perfektní a nebo, že nemířily přesně na kameru, a objevil se kvůli tomu problém s pozicí dlaní nebo hlavy, které najednou začaly vybočovat do podivných úhlů, proto je velmi důležité jejich správnou kalibraci hlídat a stejně tak jejich zamíření na senzor.

Následně jsem se vrátil do záložky Tracking a aplikoval Jitter Removal, většinou jsem nechal výchozí hodnoty a získal dobré výsledky. Tím bylo zpracování sekvence hotové a animaci jsem mohl exportovat nebo namapovat na Rig pro jiný model, čemuž se budu věnovat v jiné kapitole.

5.13.2 Sekvence s více Kinecty

Proces zpracování sekvencí s více Kinecty je v podstatě identický, proto zde poukážu pouze na ty části, které jsou odlišné.



Obrázek 69: iPi Mocap Studio - vytvoření projektu s více Kinecty

Při vytvoření nového projektu vyskočí okno, kde je nutné nejdřív zvolit, zda se bude jednat o kalibrační projekt nebo zpracování sekvence s akcí, tento výběr u souboru obsahující data z jednoho senzoru není, protože tam není co kalibrovat. Dále je nutné najít a načíst dříve zpracovanou kalibrační scénu a zvolit počet herců ve scéně (viz. obrázek č.69).

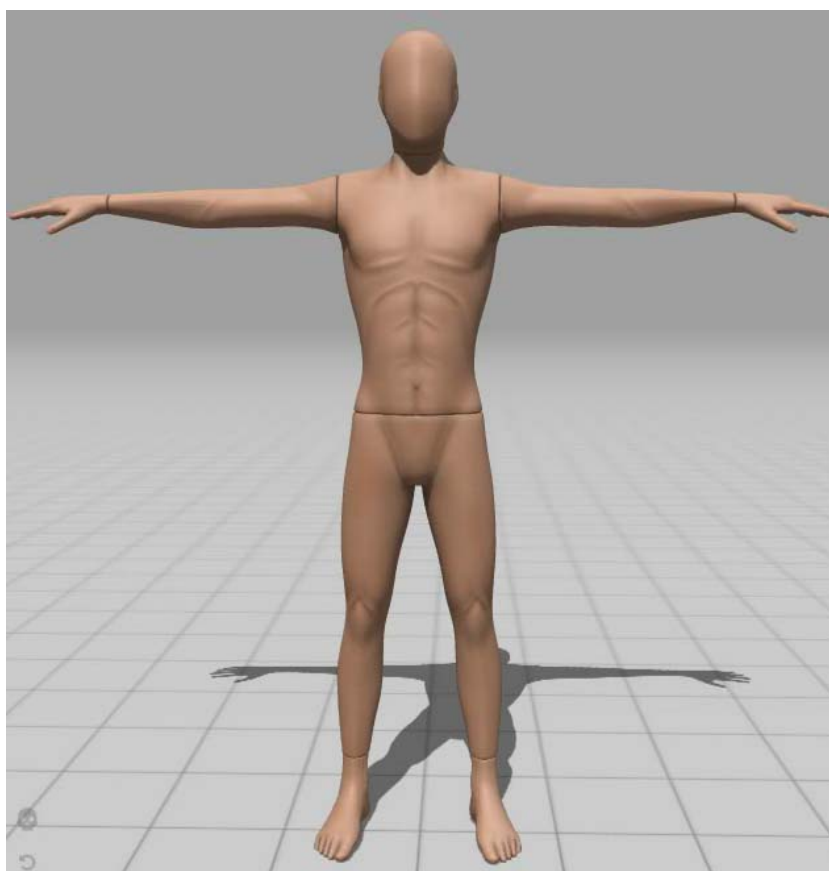
Po vytvoření projektu je možné si zapnout zobrazení hloubkových dat ze všech senzorů a měly by díky vybranému kalibračnímu projektu přesně složit hloubkový obraz herce. Jinak je celý proces prakticky identický jako u jednoho senzoru.

5.14 Export animace do Blenderu

Zpracované sekvence nejsou v iPi Mocap Studiu na jeho virtuálním herci zrovna dobře zobrazovaná pro zhodnocení, celý model herce je jen jedna jednobarevná textura a není tam žádná možnost úpravy samotné scény ani osvětlení a kvůli tomu může lidskému oku spousta detailů uniknout. Proto jsem se rozhodl zpracované sekvence namapovat na jiný humanoidní 3D model a exportovat animace do Blenderu, kde vytvořím jednoduchou scénu s otexturovaným modelem a vlastním osvětlením, aby vyniklo více detailů pro zhodnocení.

5.14.1 Výběr 3D modelu

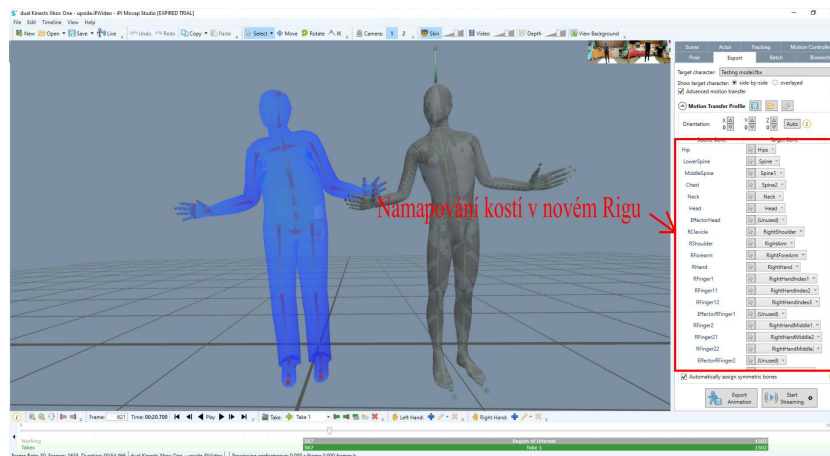
U 3D modelu jsem chtěl, aby byl relativně jednoduchý a bylo snadné pozorovat deformace těla modelu, proto jsem nechtěl využít modely, co mají na sobě několik vrstev oblečení, zbroj a další věci. Také jsem chtěl, aby byl model humanoidní, žádné přišery a podobně, aby měl stejný nebo velmi podobný rig s co možná nejbližším počtem a rozestavením kostí jako rig iPi Mocap Studia a aby byl ve formátu, který je snadné dostat do Blenderu i iPi Mocap Studia.



Obrázek 70: Model Mannequin z kolekce Mixamo

Vybral jsem si model Mannequin z kolekce Mixamo (viz. obrázek č.70), tyto modely jsou zdarma a je možné je využít pro jakýkoliv typ projektu pro komerční i nekomerční účely. Model Mannequin jsem vybral, protože je jednoduchý, nemá žádné oblečení ani jiné věci na sobě, jeho Rig poměrně přesně odpovídá Rigu Mocap Studia a je možné ho stáhnout v několika formátech vhodných jak pro Blender, tak iPi Mocap Studio.

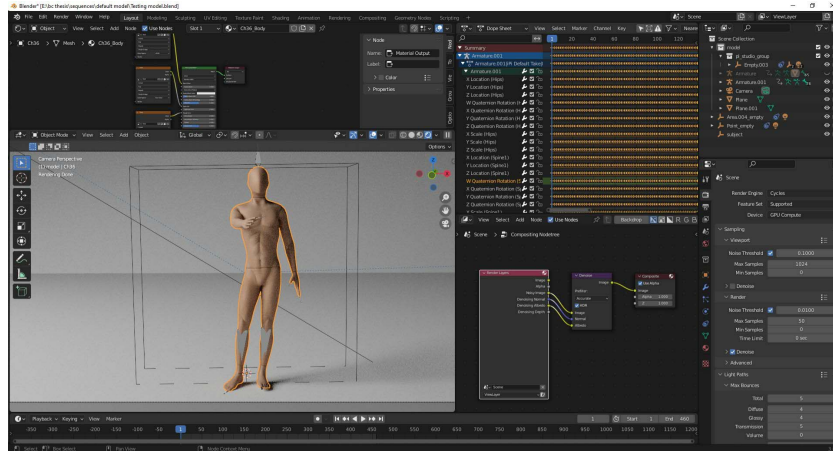
5.14.2 Namapování na nový rig a export animace



Obrázek 71: Mapování kostí nového Rigu

V záložce Export v iPi Mocap Studiu jsem v seznamu Target Character vybral možnost Import From File a vybral model Mannequin ve formátu FBX. Nastavil zobrazení nového modelu vedle originálního virtuálního herce a začal mapovat jednotlivé kosti původního Rigu na nový Rig (viz. obrázek č.71). Když byly všechny kosti správně přiřazené, animaci jsem si spustil, abych se přesvědčil, že přenos funguje dobře, a poté animaci exportoval ve formátu FBX.

5.14.3 Vytvoření scény v Blenderu a export videa



Obrázek 72: Získané sekvence v prostředí Blenderu

Do Blenderu jsem importoval model Mannequin, vytvořil dvě plochy se základním bílým materiálem a jednu nechal jako podlahu a druhou otočil o 90° a vytvořil tak zadní stěnu za modelem, na modelu jsem vytvořil materiál pro celé tělo a s pomocí textur u FBX modelu ho otexturoval. Poté jsem vytvořil jednoduché vhodné osvětlení a kameru, kterou jsem napozicoval tak, aby byl model na středu (viz. obrázek č.72).

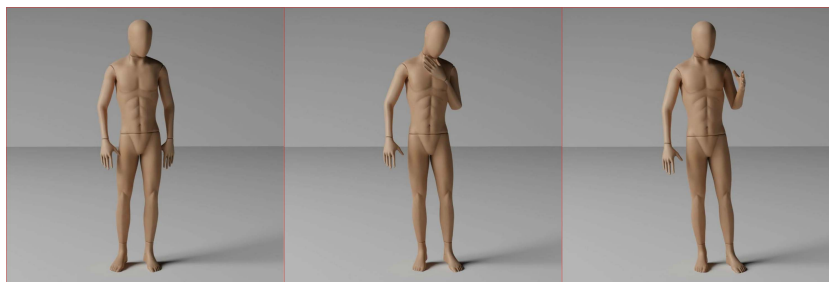
Na model jsem poté začal postupně aplikovat všechny vyexportované animace a vyrenderoval je jako videa.

6 Kvalita snímání

Všechny porovnávací a speciální sekvence lze nalézt v příloze A, na přiloženém CD, ve formátu mp4.

6.1 Minimalistický pohyb

6.1.1 Jeden Kinect v1



Obrázek 73: Minimalistický pohyb - jeden Kinect v1

Pohyb těla mi přišel poměrně přesný, i když možná trochu trhaný v určitých částech, pohyb dlaně představující ruku s cigaretou byl docela nepřesný, pozice dlaně byla mimo i o několik centimetrů a vypadala, že místo kouření cigarety si herec dlaní podpírá bradu.

6.1.2 Dva Kinecty v1

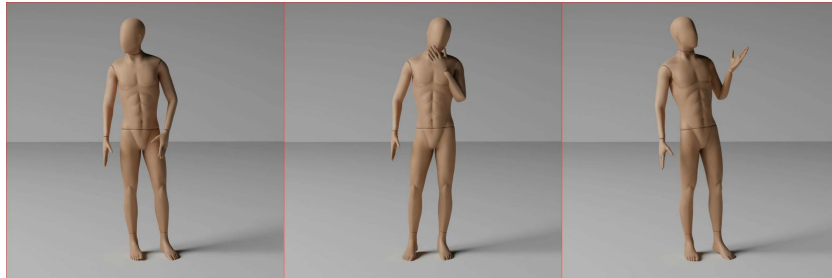


Obrázek 74: Minimalistický pohyb - dva Kinecty v1

Pohyb těla byl poměrně přesný, opět se projevilo mírné trhání. Pohyb dlaně byl také relativně přesný, dlaň se nepropadala do modelu herce a odchylka její pozice

byla mnohem menší než při využití jednoho v1 senzoru.

6.1.3 Jeden Kinect v2



Obrázek 75: Minimalistický pohyb - jeden Kinect v2

Pohyb těla byl celkem přesný, opět se objevovalo mírné trhání v určitých částech. Pohyb dlaní byl také poměrně přesný. Snímání s jedním Kinectem v2 mi přišlo stejně kvalitní jako se dvěma Kinecty v1.

6.1.4 Dva Kinecty v2



Obrázek 76: Minimalistický pohyb - dva Kinecty v2

Pohyb těla celkem přesný, s mírným občasným trháním. Pohyb dlaní byl možná o něco málo přesnější než jedním v2 senzorem. Došlo k propadnutí prstů do tváře, což ale mohlo být způsobené, protože pohyb prstů se nesnímá, a vzhledem k tomu, že dlaň byla přímo u tváře při předstíraném kouření, je to celkem pochopitelné. Vyskytla se chyba se sledováním levého chodidla, které se zaseklo v nepřírodném úhlu, a automatika to nedokázala srovnat, to se stalo nejspíš proto, že jsem dal chodidla úplně k sobě a pohupoval se na nich a ještě jsem při snímání nenosil boty.

6.1.5 Celkové zhodnocení minimalistického pohybu

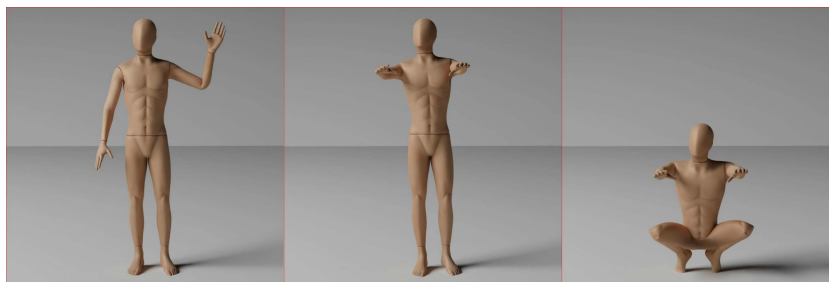
V podstatě všechny konfigurace přinesly použitelné výsledky, které by při minimálních manuálních úpravách byly použitelné.

Nejhorších výsledků dosáhl jeden Kinect v1, který měl poměrně velkou odchylku od přesné pozice dlaně. Nejlepší výsledky přineslo snímání se dvěma Kinecty v1 a jedním Kinectem v2. Snímání se dvěma Kinecty v2 sice mělo chyby, ale těm se podle mě dalo předejít, a mělo by být zařazeno jako minimálně stejně kvalitní jako jeden Kinect v2 a dva Kinecty v1.

Pro snímání minimalistického pohybu bych doporučil jakoukoli konfiguraci kromě samotného Kinect v1.

6.2 Normální pohyb

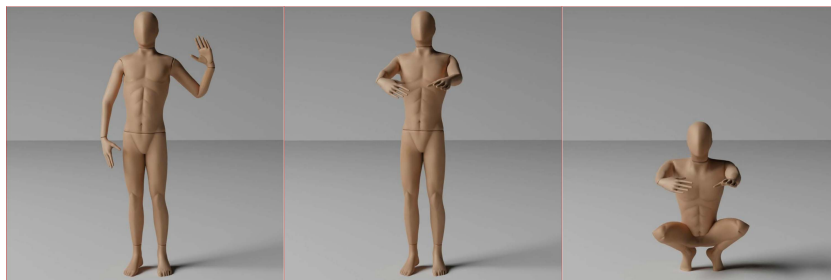
6.2.1 Jeden Kinect v1



Obrázek 77: Normální pohyb - jeden Kinect v1

Pohyb těla je poměrně přesný s mírným občasným trháním, dobrý je i pohyb dlaní a nohou, při dřepu se objevila podivná nepřírozená deformace modelu v oblasti kolen, a také část chodidel propadla pod podlahu.

6.2.2 Dva Kinecty v1



Obrázek 78: Normální pohyb - dva Kinecty v1

Kvalita byla v podstatě shodná jako s jedním Kinectem v1 i se stejnými problémy. Trhání těla mi přišlo o něco menší, až téměř neexistující, a objevil se problém se správnou rotací pravé dlaně, to ale nejspíš způsobil driftující motion controller.

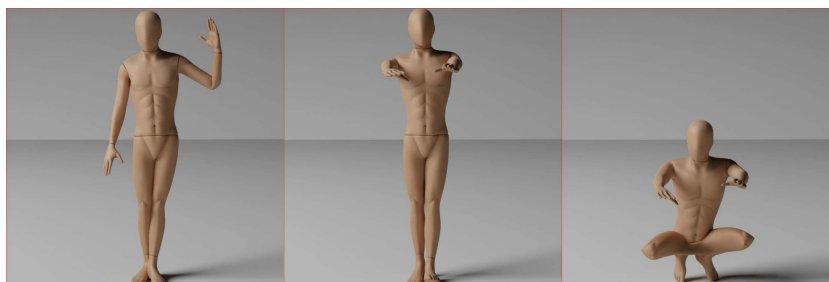
6.2.3 Jeden Kinect v2



Obrázek 79: Normální pohyb - jeden Kinect v2

Pohyb těla, dlaní i nohou se zdál velmi přesný a plynulý, bez trhání. Opět se objevil problém s propadem chodidel pod podlahu při dřepu a deformace v oblasti kolen.

6.2.4 Dva Kinecty v2



Obrázek 80: Normální pohyb - dva Kinecty v2

První část sekvence s máváním je velmi dobrá a plynulá, bez chyb, až na pozici levého chodidla, které je o zhruba o pět centimetrů blíž k pravému chodidlu, než by mělo. Při dřepu se opět objevila divná deformace v oblasti kolen, ale pozice chodidel byla na rozdíl od předchozích pokusů velmi dobrá, jen s minimálním propadem prstů pod podlahu. Také mi přijde trochu nepřírozená poloha pravé ruky při dřepu.

6.2.5 Celkové zhodnocení normálního pohybu

Celkově si všechny konfigurace s tímto pohybem poradily poměrně dobře, Kinecty v1 měly o něco větší potíže s trhaným pohybem těla. Nejlíp si vedla konfigurace dvou Kinectů v2, kdy byl pohyb velmi kvalitně zpracovaný a plynulý, chyby, které se v této sekvenci objevily, nebyly zaviněny špatným snímáním Kinect senzorů, ale driftujícími motion controllerů a absencí obuvi.

Pro snímání klasických jednoduchých pohybů bych doporučil jednu z konfigurací s Kinecty v2.

6.3 Rychlý pohyb

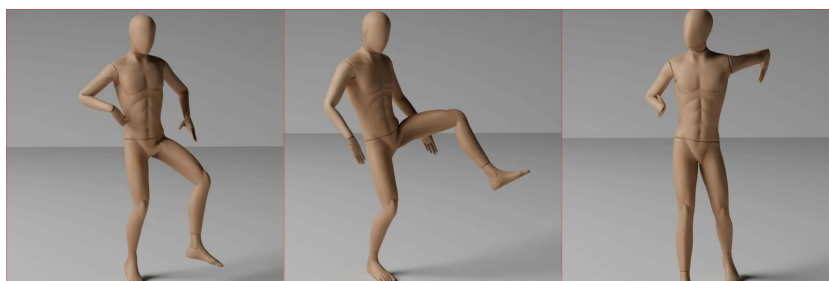
6.3.1 Jeden Kinect v1



Obrázek 81: Rychlý pohyb - jeden Kinect v1

Celkově je pohyb poměrně nepřesný, snímání horních i dolních končetin několikrát zcela vypadlo, a i když fungovalo, bylo velmi nepřesné s odchylkou i několik centimetrů, sotva bylo rozpoznat pohyb z originálního videa. Tuto sekvenci hodnotím jako nepoužitelnou i pro manuální korekci, bylo by možná snazší ji udělat manuálně celou od začátku.

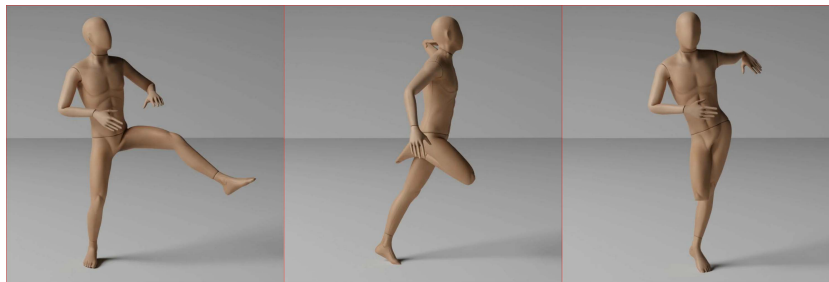
6.3.2 Dva Kinecty v1



Obrázek 82: Rychlý pohyb - dva Kinecty v1

Oproti jednomu senzoru v1 je pohyb mnohem lepší, končetiny vypadly ze snímání jen zřídka a celkově byly pozice všech částí těla přesnější, i když stále bylo poznat, že jsou o něco málo mimo. Výsledná sekvence i tak byla v poměrně špatné kvalitě a osobně bych jí použít asi nechtěl.

6.3.3 Jeden Kinect v2



Obrázek 83: Rychlý pohyb - jeden Kinect v2

Když tracking fungoval, byl celkem přesný a plynulý. Během snímání bohužel dost často vypadal tracking jednotlivých končetin.

6.3.4 Dva Kinecty v2



Obrázek 84: Rychlý pohyb - dva Kinecty v2

Kvůli rychlosti pohybu nebyl tracking vůbec přesný a objevovala se poměrně velká odchylka přinejmenším několika centimetrů od reálné pozice končetin v prostoru, nicméně byl pohyb jinak docela plynulý a za celou dobu žádná z končetin nejspíš nevytáhla z trackingu.

6.3.5 Celkové zhodnocení rychlého pohybu

Žádná konfigurace si nevedla příliš dobře a myslím si, že Kinecty zkrátka nejsou na snímání rychlých pohybů celkově vhodné kvůli jejich omezení na 30 snímků za vteřinu.

Pro snímání rychlého pohybu bych Kinecty vůbec nedoporučil, ale pokud už není zbytí, vybral bych konfiguraci se dvěma Kinecty v2.

6.4 Otáčení

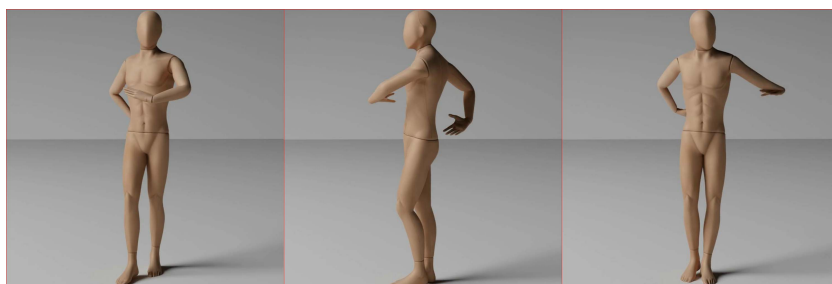
6.4.1 Jeden Kinect v1



Obrázek 85: Otáčení - jeden Kinect v1

Během otáčení byl tracking překvapivě celkem kvalitní, hlavně co se týče těla a nohou. Při kompletním překrytí části dlaní tělem tracking této části vypadl a končetina začala náhodně lítat v prostoru. Doba, kdy byla nějaká celá část končetiny zakrytá, byla velmi krátká a program ji nejspíš proto dokázal poměrně rychle namapovat zpátky na správné místo.

6.4.2 Dva Kinecty v1

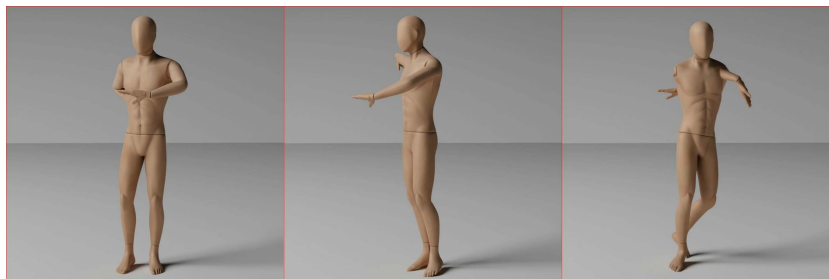


Obrázek 86: Otáčení - dva Kinecty v1

Na začátku sekvence probíhal tracking dobře, byl plynulý a na všech částech herce docela přesný. Problém byl, když v jedné části, nejspíš kvůli vysokému šumu a

nízkému rozlišení Kinectu v1, vypadlo trackování jedné ruky a automaticke se nepovedlo ji vrátit zpátky na správné místo.

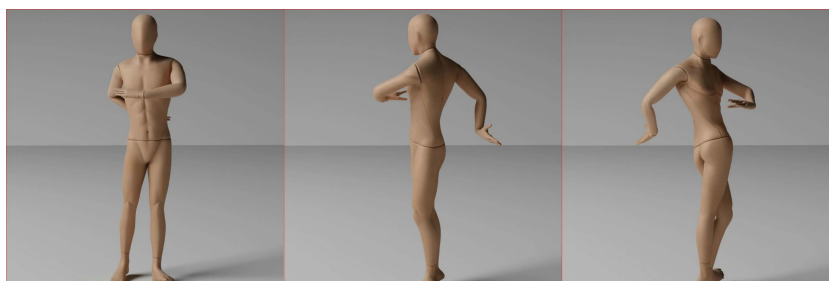
6.4.3 Jeden Kinect v2



Obrázek 87: Otáčení - jeden Kinect v2

Pozice všech částí těla byly bez větších odchylek, ovšem jakmile byla větší část končetiny zakryta, její tracking vypadl a automatika už ho nedokázala srovnat. Na konci sekvence byla jen jedna končetina stále na správném místě.

6.4.4 Dva Kinecty v2



Obrázek 88: Otáčení - dva Kinecty v2

Sekvence byla velmi plynulá a trackovaná na téměř přesných pozicích. Problém byl s trackingem pozice chodidel, což mohlo být způsobeno absencí obuvi při nahrávání sekvence. Pravá ruka byla celou dobu ohlá lehce v nepřírozeném úhlu, což ale bylo nejspíš způsobeno driftováním v motion controlleru. Celkem byl pohyb velmi dobře zachycen a žádná končetina ani jednou zcela nevypadla ani při rychlejších otáčení.

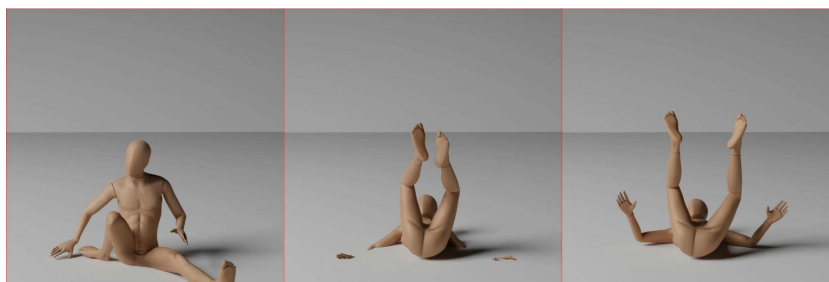
6.4.5 Celkové zhodnocení otáčení

Je jasné, že snímání s jedním senzorem bude mít vždy problémy při otáčení herce, protože když se zakryje některá část těla, data o ní zkrátka budou senzoru úplně chybět. Nejhůř si zde vedl asi samotný Kinect v2, kde vypadly z trackingu téměř všechny končetiny, myslím si, že v1 si vedl lépe, ale pouze proto, že v jeho sekvenci mizely části končetin zcela jen na velmi krátkou dobu. Nejlepší pohyb při otáčení zpracovaly dva Kinecty v2, i když se tam objevily problémy se snímáním chodidel, což podle mě způsobila absence obuvi a také divný úhel pravé dlaně, za což ale téměř určitě mohly pouze nekalibrované motion controllery.

Pro snímání pohybu, který obsahuje otáčení herce, bych doporučil jakoukoli konfiguraci s alespoň dvěma Kinecty.

6.5 Vzhůru nohama

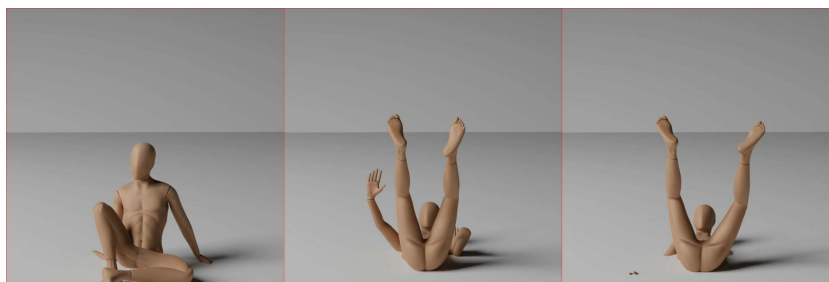
6.5.1 Jeden Kinect v1



Obrázek 89: Vzhůru nohama - jeden Kinect v1

Měl jsem problém vejít se celý do záběru při lehání na zem, proto zřejmě několikrát vypadlo snímání chodidel. V sekvenci je poměrně velké množství nepřesností a jak torzo, tak ruce propadají pod podlahu a jsou až o několik centimetrů ve špatné pozici. Jinak ale senzor neměl problém s rozpoznáním rukou a nohou i při prohození jejich pozice, kdy chodidla byla nad úrovní dlaní.

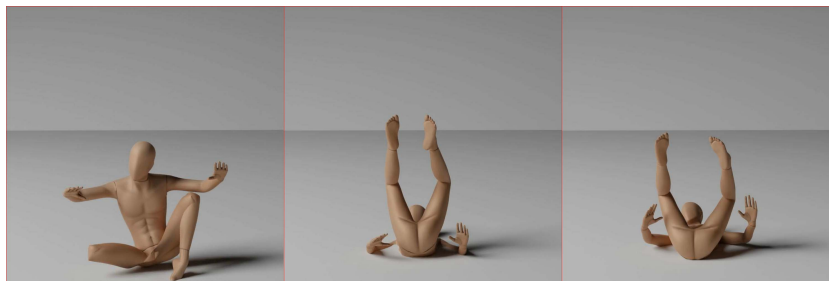
6.5.2 Dva Kinecty v1



Obrázek 90: Vzhůru nohama - dva Kinecty v1

Měl jsem opravdu velký problém vměstnat se do ještě omezenějšího snímacího prostoru než u jednoho senzoru v1. Končetiny neustále různě vypadávaly a poté se zpětně zkoušely namapovat, všechny se nakonec namapovaly na správné místo, takže senzory neměly problém s prohozením výšky chodidel a dlaní. Opět se objevovaly značné problémy s propadáváním dlaní a torza pod podlahu.

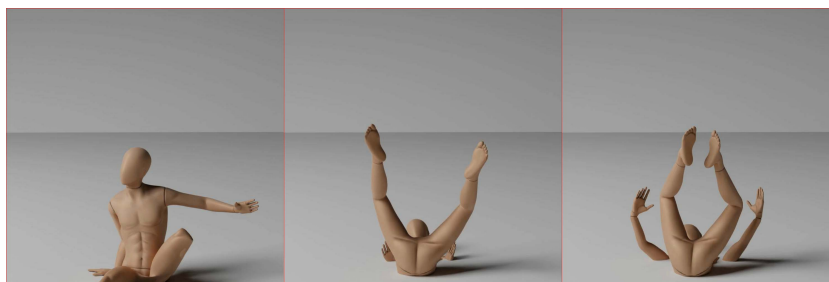
6.5.3 Jeden Kinect v2



Obrázek 91: Vzhůru nohama - jeden Kinect v2

Stejné problémy jako při využití jednoho Kinectu v1 s rozdílem, že pohyb byl o něco přesnější po většinu sekvence, naopak propad torza a dlaní pod podlahu byl zde mnohem markantnější než u senzoru v1. Zlepšením bylo, že senzor dokázal mnohem přesněji rozpoznat jednotlivé končetiny, i když se mezi sebou různě překrývaly, aniž by vypadlo jejich trackování. Senzor neměl problém s prohozením pozice dlaní a chodidel.

6.5.4 Dva Kinecty v2



Obrázek 92: Vzhůru nohama - dva Kinecty v2

Opět jsem měl velké problémy vměstnat se celý do snímací plochy a s vypadáváním snímání končetin při lehání. Propad torza a rukou pod podlahu byl zde zřejmě největší, ovšem kvalita snímání končetin byla poměrně dost slušná a po lehnutí už nedocházelo k výpadkům trackingu, i když ruce v jedné části projely skrz většinu stehna.

6.5.5 Celkové zhodnocení snímání vzhůru nohama

Senzory rozhodně nemají problém s prohozením pozice dlaní a chodidel, což jsem chtěl při snímání tohoto pohybu zjistit. Myslím si, že program, i pokud je zapnutá možnost kolizí s podlahou, hlídá jen pozici chodidel, aby pod ní nepropadla a zbytek těla nehlídá.

Všechny konfigurace měly problém s vměstnáním celého herce do snímacího prostoru při lehání, s výpadky snímání končetin i propadáváním torza a rukou pod podlahu. Nejsem si jistý, jakou konfiguraci zde vybrat jako nejlepší, ale asi bych zvolil snímání jedním Kinectem v2, protože se dvěma je snímací prostor menší, a jsou ještě větší problémy udržet celého herce ve snímacím prostoru.

Pro snímání tohoto pohybu nemohu zcela doporučit žádnou konfiguraci, protože se ve všech vyskytly v podstatě stejné problémy. Nejlépe si asi vedly sekvence s Kinecty v2, i když se v nich objevoval větší propad torza pod podlahu.

6.6 Porovnání jednoho a dvou Kinectů

Když srovnám, jak si ve všech sekvencích vedly konfigurace s jedním a se dvěma senzory, dospěl jsem k názoru, že není zcela jasný stoprocentní vítěz.

6.6.1 Snímání s jedním Kinectem

Snímání s jedním senzorem má své výhody, jednodušší sestavení, není potřeba přemýšlet nad rozmístěním sensorů, není potřeba řešit kalibraci sensorů, je k dispozici o něco větší snímací plocha a ve většině případů si vedou relativně obstojně.

Nevýhody snímání s jedním senzorem jsou patrné při jakékoliv scéně, kde je potřeba komplexnější pohyb herce, který se potřebuje v prostoru otáčet, v tu chvíli začne docházet k problémům s trackingem končetin.

6.6.2 Snímání s více Kinecty

Nevýhody jsou složitější sestavení, nutné správné rozmístění sensorů, nutnost kalibrace, menší snímací plocha.

Výhoda je možnost snímat i mnohem komplexnější pohyby, celé tělo herce je snímáno z více obou stran, takže je menší šance, že se vyskytne chyba ve snímání, a když je některá část těla skrytá jednomu senzoru, nejspíš jí bude snímat alespoň druhý senzor. Pozice končetin je většinou přesnější s menší odchylkou a celková kvalita snímání je rozpoznatelně lepší, i když popravdě podle mě ne zase tak o moc.

6.6.3 Zhodnocení

Konfigurace se dvěma senzory bych doporučil komukoliv, kdo chce snímat komplexnější pohyby a nebýt limitovaný tím, že se herec nemůže pořádně otočit a také pokud mu záleží na o něco málo větší přesnosti pozicování. Pokud uživateli stačí snímání jednoduchých pohybů, jeden Kinect bude dostačující.

S minimalistickým a normálním pohybem si dokázaly obě konfigurace poradit dobře, co se týče snímání vzhůru nohama, vždy se projevil problém s propadnutím těla a rukou podlahou a rychlé snímání bylo problematické v každé konfiguraci,

kvůli omezeným snímkům za vteřinu. Při určité rychlosti pohybu je zkrátka rozmazání pohybem tak vysoké, že to Kinecty nedokážou zřejmě zpracovat.

6.7 Porovnání snímání Kinectů v1 a v2

Větší rozlišení Kinectů v2 a jejich větší zorné pole umožňuje o něco přesnější snímání na větší ploše, ve většině sekvencí s Kinecty v2 mi přišlo, že pozice končetin a těla je přesnější než s Kinecty v1. Na druhou stranu mi přišlo, že v několika sekvencích tracking s Kinecty v2 je náchylnější na selhání sledování, pokud nějaká část těla zmizela na chvíli z viditelného pole, obzvláště pokud nebyla kalibrace perfektní.

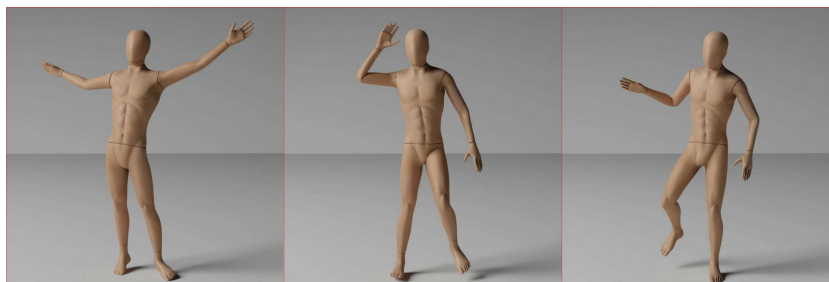
Celkově ale hodnotím Kinecty v2 jako lepší volbu ke snímání, pokud si uživatel dá tu práci je správně nastavit a perfektně zkalibrovat.

6.8 Vliv velikosti herce

Zajímalo mě, zda hraje nějakou roli velikost herce při snímání. Nasnímal jsem tedy jednu sekvenci s pomocí svého dvouletého syna jako figuranta a zpracoval ji v iPi Mocap Studiu.



Obrázek 93: Velikost herce - rozdílný poměr velikosti hlavy k tělu u dítěte a dospělého

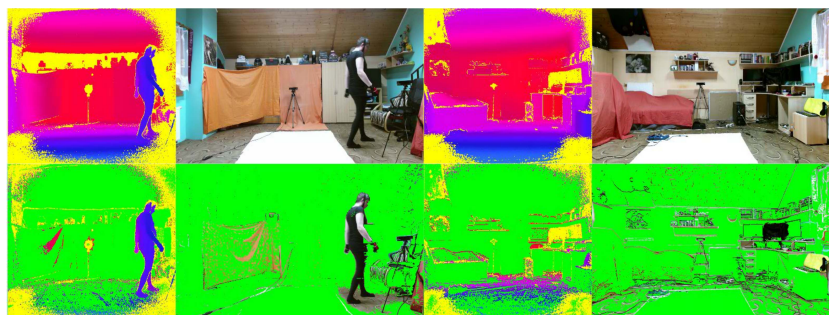


Obrázek 94: Snímání malé osoby - zpracovaná sekvence

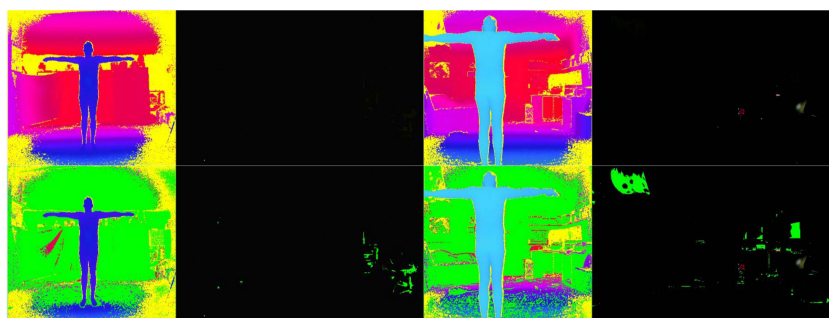
Zjistil jsem, že výška velkou roli nehraje, ale také že poměr velikosti hlavy a těla je u dětí problematický faktor pro Mo-cap v iPi Mocap Studiu. Děti zřejmě běžně mají mnohem větší hlavy v poměru k tělu než u dospělých a s tím iPi Mocap Studio nepočítá. Při zmenšení velikosti virtuálního herce na velikost dítěte, je hlava mnohem menší než u dítěte a není možné to nijak kompenzovat (viz. obrázek č.93), takže není možné postavu herce správně zarovnat s hloubkovými daty. Nicméně tracking těla bylo možné provést a fungoval poměrně dobře.

6.9 Možnost snímání ve tmě

Zajímalo mě, zda je možné snímat s Kinecty i v naprosté tmě, a tak jsem v noci nahrál sekvenci s vypnutým osvětlením. Sekvenci bylo možné bez problémů následně zpracovat v iPi Mocap Studiu, takže je možné nahrávat s Kinecty i v naprosté tmě při využití dat pouze z hloubkových kamer.

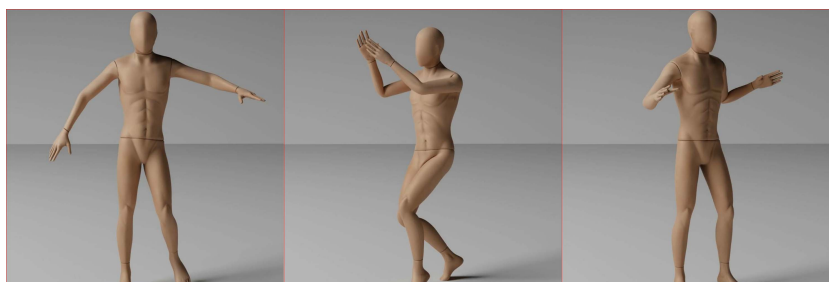


Obrázek 95: Snímání ve tmě - klíčování scény za světla



Obrázek 96: Snímání ve tmě - klíčování stejné scény po zhasnutí světla

Pokud bych ale použil klasické RGB kamery, snímání ve tmě by nebylo možné, jak je vidět na obrázku č.95 a obrázku č.96, vypnutí světla nemělo na hloubkové senzory a jejich klíčování pozadí žádný účinek, ale klíčování pozadí u RGB kamer v podstatě úplně přestalo fungovat.



Obrázek 97: Snímání ve tmě - zpracovaná sekvence

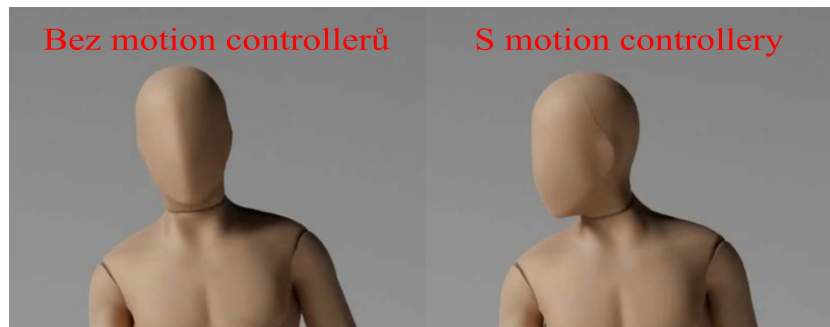
6.10 Zlepšení trackingu chodidel pomocí bot

Poměrně často se objevoval problém se sledováním pozice a hlavně rotace chodidel. Tento problém jsem se pokusil vyřešit nošením bot při snímání, protože jsem si myslel, že tím, jak boty zvětší prostor zabíraný chodidly, Kinecty budou moci lépe odlišit je od podlahy a sledovat jejich pohyb a rotaci. Myslím, že tato metoda alespoň částečně fungovala a tracking chodidel mi přišel o něco málo lepší.

6.11 Využití motion controllerů

Pro zjištění, jak velký vliv mají data z motion controllerů, jsem jednoduše v jedné sekvenci exportoval animaci bez aplikování dat z motion controllerů a poté jejich

data aplikoval a animaci exportoval znovu.



Obrázek 98: Využití motion controllerů k rotaci hlavy

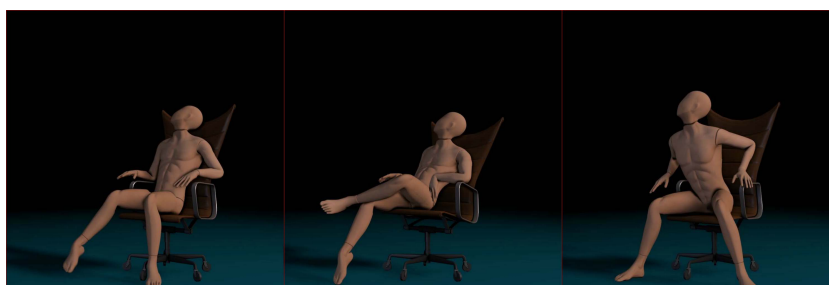
Mé výsledky poměrně jasně ukázaly, že při správné kalibraci se výrazně zlepšuje jak rotace dlaní, tak hlavy.

6.12 Interakce s objekty

Zajímalo mě, je-li možné interagovat při snímání s rekvizitami a napodobit například akce jako posazení a točení se na židli a zvednutí a manipulace s nějakým malým předmětem.

6.12.1 Sezení na židli

Sekvenci jsem se nejdříve pokusil nasnímat s jedním senzorem a výsledky byly nepoužitelné, tak jsem to zkusil se dvěma Kinect senzory v2. Po evaluaci pozadí jsem do scény umístil židli, se kterou budu interagovat. Neměl jsem ji tam na evaluaci pozadí, protože s ní budu pracovat ve scéně a nebude statickým objektem, což by nejspíš vadilo, pokud bych ji tam při evaluaci měl. Při nahrávání jsem došel k židli a posadil se na ní, poté se s ní zkusil po scéně přesouvat a točit se v ní.



Obrázek 99: Sekvence sezení v židli

V iPi Mocap Studiu jsem se k mému překvapení se dvěma senzory při trackingu dostal k poměrně dobrým výsledkům. Animaci jsem exportoval do Source Filmmakeru (SFM) ve formátu DMX a aplikoval ji na model Mannequin, do scény jsem pro lepší představu umístil objekt židle z kolekce modelů, kterou jsem v programu měl připravenou, a napozicoval ji tak, aby do ní model dosedl a následně manuálně zhruba naanimoval její pohyb podle pohybu hercova těla v ní. Výsledná hrubá animace nebyla vůbec špatná, bylo by potřeba udělat poměrně velkou řadu korekcí, hlavně co se týče rukou a propadání pozadí spodkem židle, ale dostal jsem tímto způsobem rozhodně poměrně kvalitní základ k dalšímu zpracování.

Stejnou sekvenci jsem se pokusil nasnímat také se třemi Kinect senzory a v iPi Mocap Studiu dostal téměř shodné výsledky jako při použití dvou. Přišlo mi, že tracking rukou byl při použití tří senzorů o něco přesnější.

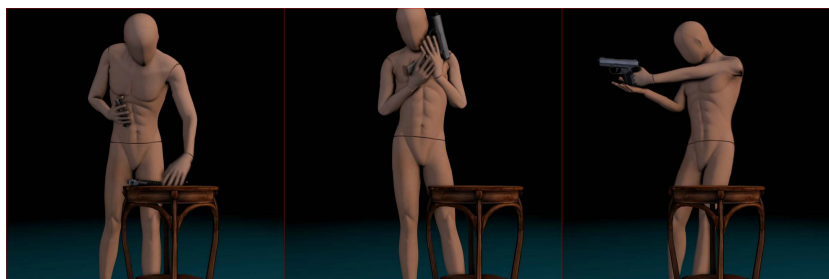
Podle mého názoru je možné nasnímat kvalitní základ animace pro sezení a s trochou manuální korekce získat kvalitní animaci.

6.12.2 Interakce s malým předmětem

Pro interakci s malým předmětem jsem si vybral poměrně složitou scénu. Herec dojde ke stolu, na kterém leží zbraň, zásobník a náboje, náboj vloží do zásobníku, ten poté do pistole, pistole se natáhne a zamíří vpřed, poté si s ní herec pohazuje a položí ji zpátky na stůl.

Do scény jsem po evaluaci pozadí umístil počítačovou skříň a na ní pistoli, náboj a zásobník. Sekvenci nahrál se dvěma senzory a šel ji zpracovat do iPi Mocap Studia. Samotná hloubková data byla velmi dobrá, ale protože v této scéně hraje velkou roli pohyb a ohýbání prstů a rotace dlaní, tracking nepřinesl úplně

kvalitní výsledky, tracking prstů není v iPi Mocap Studiu možný a je třeba ho udělat manuálně, což je velice zdoluhavý proces pro delší sekvence a s přesnou rotací dlaní má také problémy bez využití motion controllerů.



Obrázek 100: Interakce s malým objektem

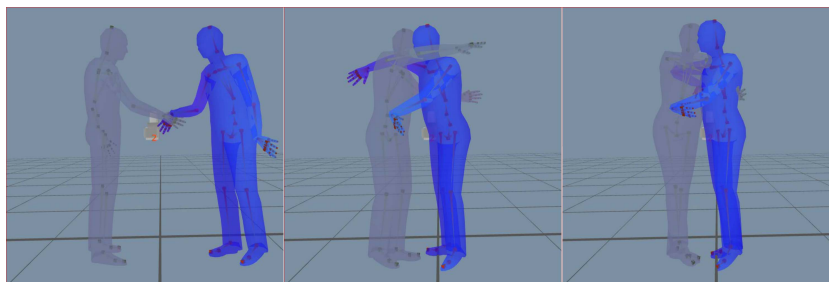
Po exportu do Source Filmmakeru jsem do scény umístil ze své připravené kolekce model stolu, pistole, zásobníku a náboje a potom postupně zamykal pohyb těchto modelů k pohybu dlaně, která s nimi interagovala, ten následně přepínal na dlaň druhou nebo další 3D model a prováděl základní manuální korekci rotace.

Výsledek nebyl nijak zvlášť dobrý, ale s vynaložením poměrně značného manuálního zásahu do pohybu rukou, prstů a dalších vylepšení je možné na tomto základu vybudovat dobrou animační sekvenci. Pokusil jsem se této sekvenci ještě chvíli věnovat a po několika úpravách se dobral k již poměrně přijatelným výsledkům, a když bych u ní strávil ještě možná dvě hodiny, animace by byla zřejmě na velice dobré úrovni.

Podle mě je možné pro manipulaci s malým objektem pomocí Mo-cap s Kinecty v iPi Studiu získat relativně dobrý základ animace, který vyžaduje značnou manuální korekci.

6.13 Snímání více herců

Již dříve jsem psal, že podle mě je pro snímání s Kinecty dělat scény s více herci nevhodné, kvůli malé snímací ploše. Nicméně jsem se ji rozhodl vyzkoušet a požádal dědu, aby mi dělal druhého herce. Na scéně nebylo dost prostoru, abychom oba mohli udělat T-pose, tak jsme ji udělali postupně a druhý herec se vždy uklidil na okraj snímané oblasti. Následně jsme udělali krátkou sekvenci, kdy jsme k sobě přišli, podali si ruce a objali se.



Obrázek 101: Interakce více herců

Při zpracování sekvence se v iPi Mocap Studiu objevili dva virtuální herci místo jednoho. Problém byl, že T-pose jsme dělali v rozdílných časech a nešlo nastavit vypnutí trackingu jednoho herce do jiného časového bodu, háček je, že vypnutí sledování určité části těla platí zřejmě globálně pro všechny virtuální herce a nejde rozdělit na jednotlivé rigy (nebo jsem přinejmenším nepřišel na to, jak to udělat), takže se druhý herec zoufale pokoušel na něco namapovat, i když neměl na co, ve chvíli, kdy jsem ho mohl namapovat na T-pose, byl jeho model příšerně deformovaný ve scéně a jeho korekce byla velmi otravná.

Finální animace nebyla k mému překvapení úplně špatná, až na propadávání dlaní do těla druhého herce a dalších drobných chyb. Stále si myslím, že není snímání s Kinecty vhodné pro více herců, uznávám ale, že na některé jednoduché interakce je možné tuto možnost použít a dosáhnout přijatelných výsledků při snímání.

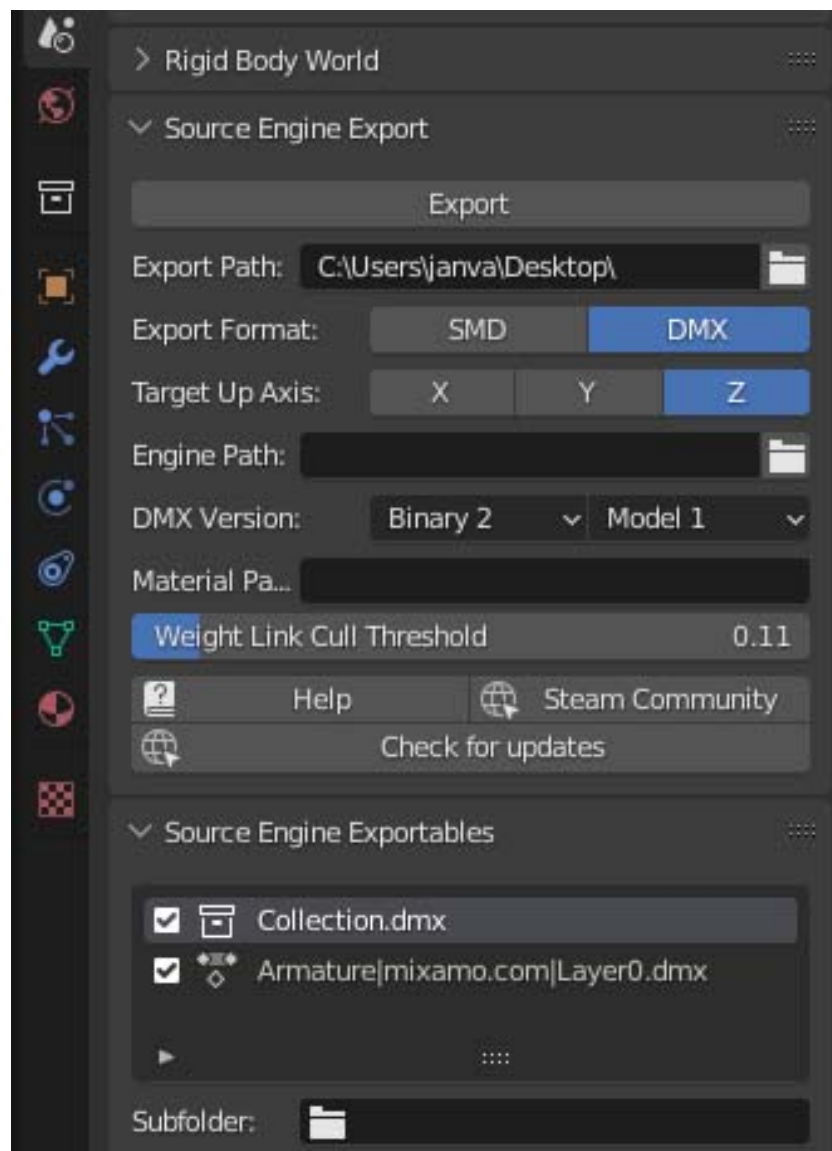
7 Zpracování v Blenderu a SFM

Po prvotním zpracování sekvencí iPi Mocap Studiem je další krok exportovat animace pro zpracování v dalších programech. V mém případě to byl Blender a Source Filmmaker.

7.1 Export modelu do SFM

Protože jsem chtěl zjistit, jak dobře oba programy dokážou zpracovat získané sekvence, musel jsem je porovnávat na stejném modelu. Proto jsem musel model Mannequin dostat z Blenderu do SFM.

SFM umí pracovat jen s modely ve formátech SMD a DMX, takže jsem ho musel do jednoho z těchto formátů konvertovat. O procesu konverze modelů z Blenderu do SFM by se nejspíš dala napsat celá další bakalářská práce, takže tuto část jen zběžně shrnu.



Obrázek 102: Source Tools Addon

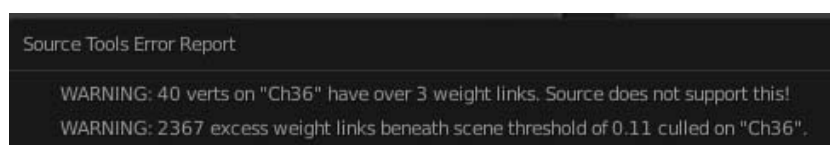
Potřeboval jsem addon k Blenderu nazvaný Blender Source Tools, který umožní importovat a exportovat modely do modelu smd a dmx, tento addon je zdarma. Stáhl jsem aktuální verzi addonu 3.2, která fungovala i s novou verzí Blenderu 3.0.0 a addon do Blenderu nainstaloval, v záložce edit - preferences - add-ons - install. V záložce scene properties se objevila nová záložka Source Engine Export (viz. obrázek č.102).

V rigu 3D modelu jsem musel přejmenovat jednotlivé kosti podle formátu, který používá SFM, aby bylo možné následně využít defaultní IK Rig v SFM pro ovládání modelu. Takže jsem Rig přepnul do Edit Mode a jednotlivé kosti v záložce Bones přejmenoval (viz.obrázek č.103). Model by měl teď fungovat se SFM scriptem `rig_biped_3_spine`, pokud ho správně zpracuji.



Obrázek 103: Kosti rigu přejmenované podle Valve Biped standartu

V okně přidaného addonu jsem nastavil export path kam jsem chtěl model vyexportovat, formát na dmx, který je novější a nahrazuje smd formát, kromě toho je jednodušší jeho zpracování, nepotřebuje speciální VTA soubor pro zpracování posuvníků upravujících části těla (flexes) a umí pracovat s několika novějšími funkcemi, které smd neumělo. Target up Axis jsem nechal jako Z, protože osa Z by měla být výšková, jsou ale modely, které to z nějakého důvodu mohou mít jinak, a pokud bych to nastavil špatně, objevovaly by se v SFM ve špatné pozici. Engine path jsem nastavil do složky se SFM, DMX version jsem nechal, jak je a material path nechal prázdnou.



Obrázek 104: Problémy s weight links na vertexech 3D modelu

Weight Link Cull Threshold jsem nechal nejdřív na nule, ale při exportu se

objevila chyba, kdy stovky vertexů měly přiřazeno víc než 3 weight links (viz. obrázek č.104), což SFM nepodporuje, takže jsem ho musel postupně nastavovat do takových hodnot, až nebyl přes limit ani jeden. To může způsobit odlišnou deformaci modelu, protože culling postupně odstraňuje z chybových vertexů weight links (které určují, jaká část rigu má jak velký vliv na jakou část modelu), které jsou k nim přiřazené.

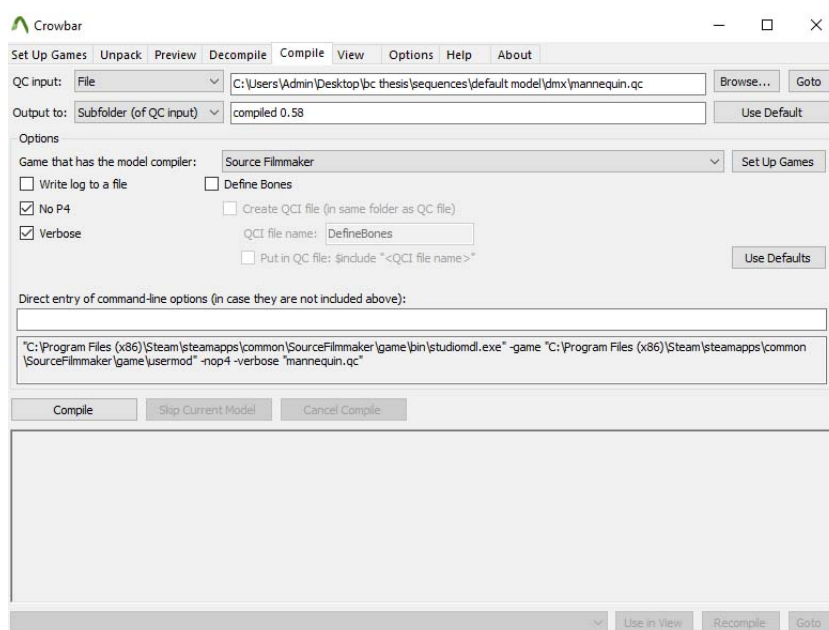
Poté jsem v Blenderu otevřel okno s jeho image editorem a postupně otevřel a vyexportoval všechny textury jako PNG obrázek. V 3D viewportu jsem vybral model Mannequin a v záložce Materials jsem si zjistil, jak se jmenují jeho materiály, protože stejně budu muset vytvořit a pojmenovat vmt soubory i pro SFM, jinak se na model materiály nenamapují.

Ve vybrané složce se teď nachází zvolené části modelu vyexportované v dmx nebo smd, tento model se skládal pouze z jednoho spojeného kusu, takže pracuji pouze s jedním dmx souborem. Pokud uživatel chce, aby z modelu šly jednoduše přepínat různé části modelu, musí se vyexportovat jako oddělené dmx soubory a při kompilaci je v kódu definovat jako přepínatelné skupiny (např. různé druhy oblečení, brýle, prsteny, klobouky atd...).

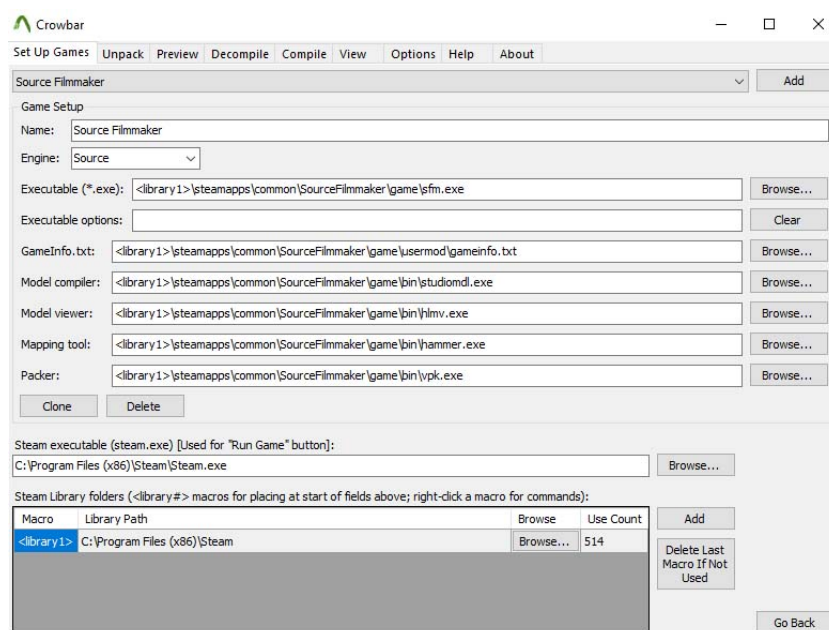
```
$ModelName "thesistest\mannequin\mannequin.mdl"  
  
$BodyGroup "body"  
{  
    studio "model.dmx"  
}  
  
$SurfaceProp "default"  
  
$Contents "solid"  
  
$EyePosition 0 0 70  
  
$MaxEyeDeflection 90  
  
$AmbientBoost  
  
$MostlyOpaque  
  
$CDMaterials "models\thesistest\mannequin\  
  
$CBox 0 0 0 0 0 0  
  
$BBox -4.147 -30.637 0 7.954 30.637 65.667  
  
$Sequence idle "model" loop ACT_IDLE 1 fps 30.00
```

Obrázek 105: Script pro kompilaci modelu Mannequin pro SFM

Model ještě musím zkompileovat pro SFM, to znamená napsání qc scriptu, je možné ho napsat v klasickém poznámkovém bloku, já použil pspad. V tomto kompilačním scriptu jsem definoval cestu k souborům modelu pro SFM, definoval části modelu, zpracování materiálů, hitboxy, cestu k souborům s materiály modelu a další věci, dá se zde například definovat několik různých druhů variant skinů modelu, přesné výchozí hodnoty jednotlivých kostí, jednotlivé flexes modelu a tak dále. Pro mé účely postačil script na obrázku č.105.



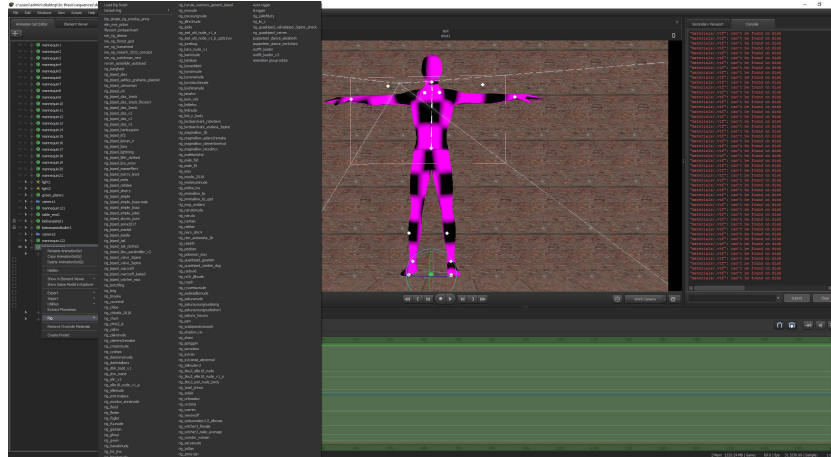
Obrázek 106: Kompilace modelu pomocí Crowbaru



Obrázek 107: Crowbar - nastavení cest programu k potřebným komponentům SFM

Ke kompilaci modelu jsem použil program Crowbar, v záložce compile jsem nastavil cestu ke svému qc souboru, kam chci zkompileovaný model vložit, pro jaký

software model kompiluji a veškeré cesty k jeho potřebným komponentům a stiskl tlačítko compile. Výsledný zkompilovaný model jsem přesunul do předdefinované cesty v qc souboru ve složce usermod a zkusil ho načíst v SFM.



Obrázek 108: Načtení modelu Mannequin do SFM bez materiálů

Model se podařilo nalézt a úspěšně načíst, aplikoval jsem na něj i IK script a model bylo možné bez problému ovládat se správnými deformacemi. Samozřejmě se načel bez materiálů (viz. obrázek č.108), protože jsem je zatím nepřipravil, tak jsem se do toho hned vrhl.

SFM mapuje materiály pomocí vmt scriptů a zpracovává textury ve vtf formátu, to znamená, že všechny textury musí být do tohoto formátu převedeny. V tomto případě jsem měl textury z Blenderu ve formátu PNG, otevřel jsem je v programu GIMP 2.0 s VTF addonem, protože GIMP neumí nativně s VTF pracovat ani do tohoto formátu exportovat (je samozřejmě možné pro konverzi použít i jiné programy jako VTFEdit). V GIMPu jsem textury jednoduše jen uložil ve formátu VTF a vložil je do složky s materiály modelu Mannequin definované v qc souboru použitým při kompilaci modelu.



Obrázek 109: Příprava textur pro SFM

Pro každý jednotlivý materiál použitý na modelu, je nutné napsat vlastní řídicí VMT script aby s ním SFM mohlo pracovat, to je u modelů s desítkami materiálů opravdu velmi nepříjemný a zdlouhavý proces, naštěstí model Manequin, který jsem vybral, měl jen jeden hlavní materiál. Ve VMT scriptu se dají definovat veškeré parametry chování materiálu, které SFM podporuje, které textury se mají použít, jak materiál reaguje na kolize se světlem, míra odlesků, hrubost povrchu, textury detailů, průhlednost a mnoho dalších. Po napsání scriptu pro materiál (viz. obrázek č.110) jsem model znovu načítal v SFM a mohl s ním začít pracovat.

Na první pohled je patrný rozdíl kvality zobrazení mezi SFM a Blenderem (viz. obrázek č.111), což je způsobeno především zastaralým renderovacím enginem, který SFM používá.

```
"VertexlitGeneric"
{
  "$basetexture" "models/thesistest/mannequin/mannequin_diff"
  "$bumpmap" "models/thesistest/mannequin/mannequin_norm"
  "$nocull" "1"

  "$phong" "1"
  "$phongexponent" 0
  "$phongboost" 0.001

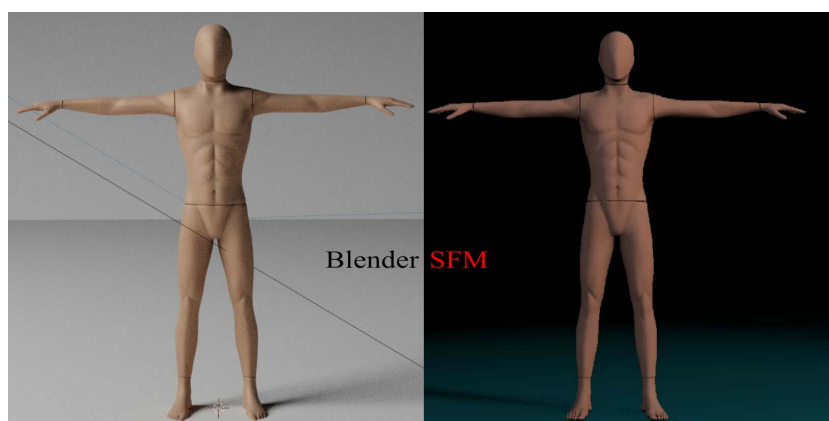
  //Phong
  "$Phong" "1"
  // "$PhongBoost" "50.15125"
  "$PhongBoost" "2.0"
  "$PhongFresnelRanges" "[0.3 0.6 0.9]"
  // "$PhongAlbedoTint" "1"
  "$PhongAlbedoTint" "0"
  "$PhongDisableHalfLambert" "1"
  "$NormalmapAlphaPhongMask" "1"

  //Rimlight
  "$Rimlight" "0"

  //Transparency
  "$AlphaTest" "1"
  "$AllowAlphaToCoverage" "0"
  "$AlphaTestReference" "0.1"

  //Other
  "$Model" "1"
  "$HalfLambert" "0"
  "$NoDecal" "1"
  "$AmbientOcclusion" "0"
  "$NoCull" "1"
  "$SurfaceProp" "flesh"
  // "$BlendTintByBaseAlpha" "1"
  "$Color2" "{ 375 355 455 }"
  "$CloakPassEnabled" "1"
  // "$CloakColorTint" "[0.4 0.5 1]"
  // "$CloakColorTint" "[1 0.5 0.4]"
}
```

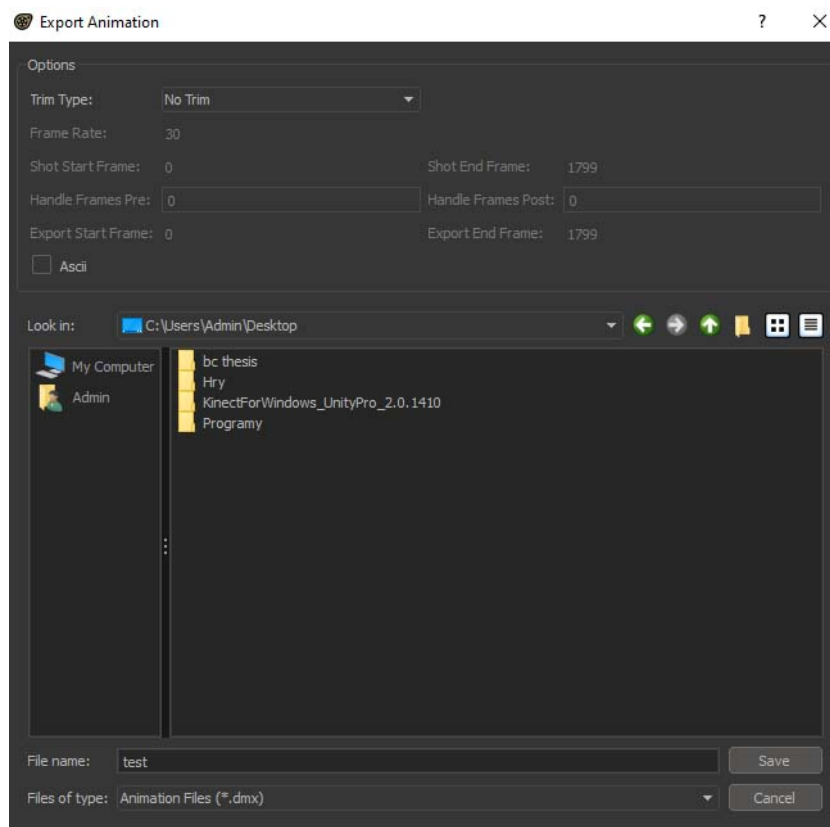
Obrázek 110: VMT script pro materiál v SFM



Obrázek 111: Model Mannequin v Blenderu a SFM

7.2 Export z iPi studia do SFM

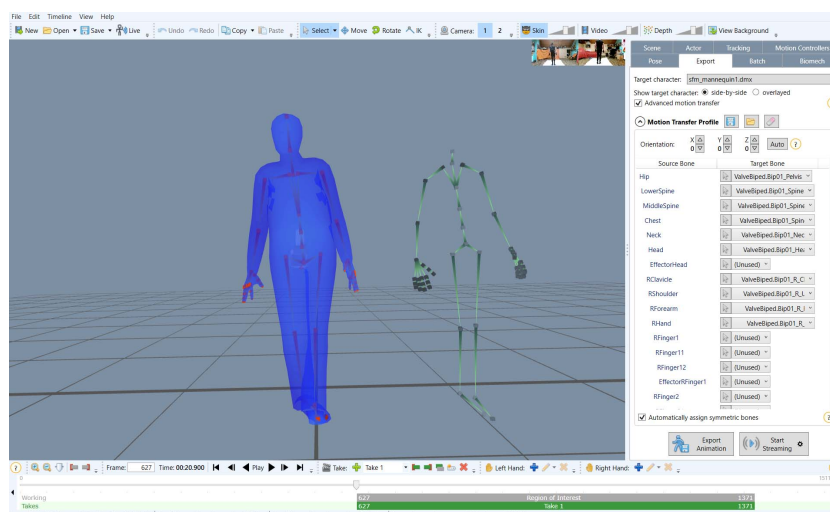
Abych mohl porovnat kvalitu zpracování sekvencí mezi Blenderem a SFM, musím nejdřív dostat získané animované sekvence do SFM. Otevřel jsem SFM a do scény přidal model Mannequin, následně jsem na jeho položku klikl pravým tlačítkem myši a vybral Export - Animation.



Obrázek 112: Export DMX pro práci v iPi Mocap Studiu

V okně, které se objevilo, jsem zvolil možnosti No Trim a ASCII, nastavil cestu pro export a jméno souboru dmx. V iPi Mocap Studiu jsem si otevřel požadovanou sekvenci a v záložce export vybral jako target character vyexportovaný dmx soubor. Potom jsem namapoval veškeré možné korespondující kosti v rigu a animaci si přehrál. Je možné, že model bude mít jiný počet kostí páteře než defaultní iPi rig, nebo rozdělené kosti v ruce pro lokaci a úpravu rotace a spoustu dalších možných odchylek, je důležité pokusit se namapovat kosti co nejpřesněji a ty, které namapovat nelze, se musí následně vyřešit manuální animací v jiných programech pro práci s 3D grafikou.

Výslednou animaci jsem vyexportoval z iPi Mocap Studia jako DMX, které jsem mohl importovat na modely v SFM.



Obrázek 113: DMX iPi Mocap Studio rig retargeting

7.3 Zpracování sekvencí v SFM

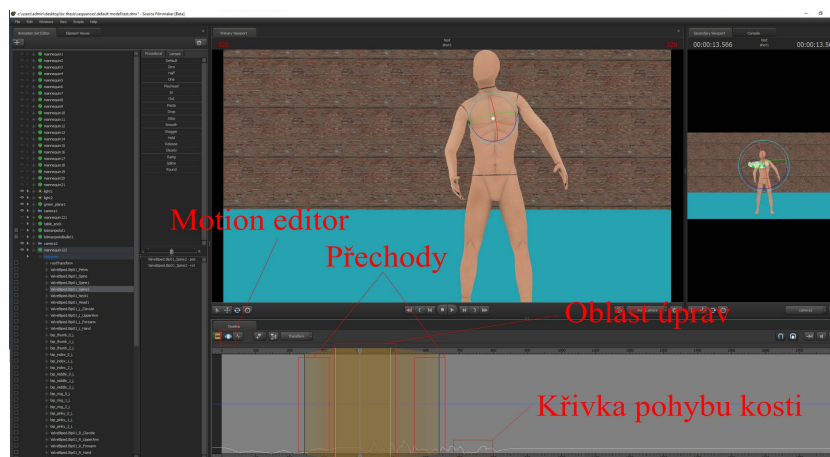
V SFM jsem vybral model mannequin, pravým tlačítkem myši, zvolil import - animation a vybral soubor dmx s animací. V SFM vyskočilo okno s mapováním jednotlivých kostí rigů ve zvoleném dmx na zvolený model (viz. obrázek č.114), vzhledem k tomu, že se jedná o totožný model, mapování proběhlo hladce. Model v SFM nyní prováděl stejný pohyb, jako v sekvenci získané z iPi Mocap Studia. To je jediná možnost jak v SFM na modely dostat nové externí animační sekvence.



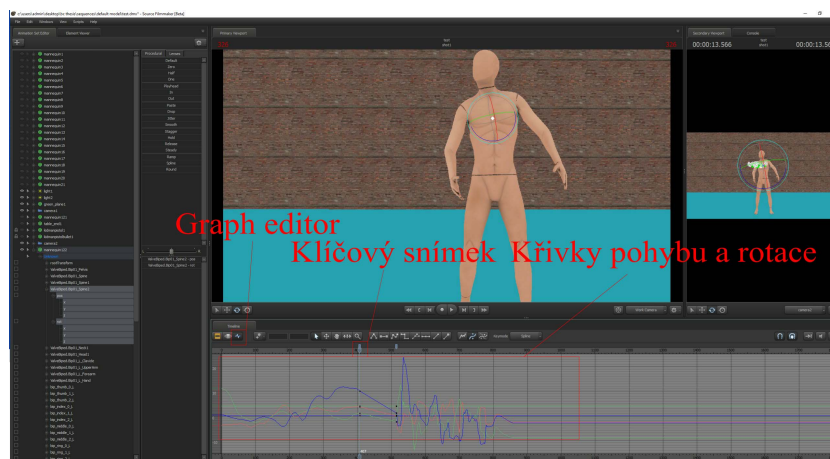
Obrázek 114: Import animace na model Mannequin v SFM

7.4 Možnosti úprav sekvence

Následná práce s importovanou sekvencí v SFM je poměrně dost omezená. Je to hlavně proto, že SFM má jen jednu animační vrstvu a jednoduše není možné vytvořit si novou prázdnou vrstvu pro korektury a kombinovat je spolu. Je do jisté míry možné udělat falešný druhotný pohyb kombinací využití motion a graph editoru, ale i tato možnost je velmi omezená. Motion editor umožňuje pro vybranou kost upravit pozici a rotaci v dané vybrané oblasti s nastavením délky přechodů a graph editor zobrazuje jednotlivé křivky pohybu a rotace, kde je možné je upravovat pomocí klíčových snímků.



Obrázek 115: SFM Motion Editor



Obrázek 116: SFM Graph Editor

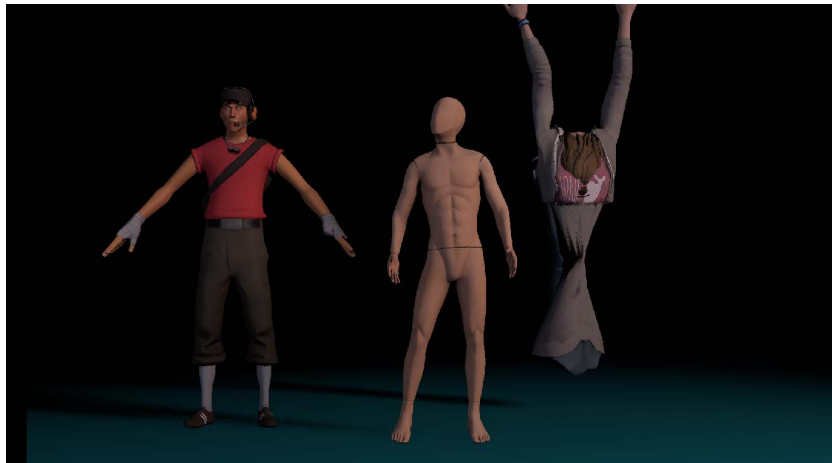
Je možné na model aplikovat IK rig, ale při úpravě pozice pomocí graph editoru se přepíše při vytvoření klíčových snímků původní nasnímaná animace pro danou kost (jak je vidět na obrázku č.114), kde mezi dvěma vytvořenými snímky došlo k novému propočítání pohybu mezi nimi a tím i zničení původní křivky, což je k ničemu. Bylo by nejspíš možné udělat klíčové snímky ve všech snímcích a postupně je odstraňovat a upravovat, což zní velmi zdlouhavě a neprakticky. V motion editoru je možné v jednotlivých částech animace upravit pozici a rotaci pro vybranou oblast a délku jejího přechodu se zachováním původních nasnímaných dat. Tato možnost je ovšem velice neohrabaná a nedá se s ní dosáhnout příliš

přesných výsledků.

Úprava získaných sekvencí je tedy možná, ovšem velmi limitovaná a za mě osobně nepraktická.

7.5 Retargeting animace

Sekvence je možné pokusit se importovat na jakýkoliv další model, ale pokud má jiné pojmenování kostí, převrácený směr jednotlivých os, jiné vztahy skupin vertexů ke kostem rigu nebo jiný počet kostí, přenos nebude dokonalý. Nemá absolutně žádný smysl se pokoušet například přenést animaci z humanoidního bipedal rigu na quadripped rig zvířete, výsledek bude pravděpodobně děsivá deformace modelu.



Obrázek 117: Import animace v SFM na různé modely

Problémy se vyskytují i při importu na humanoidní rig, který je lehce rozdílný od původního zpracovaného v dmx souboru, jak lze vidět na obrázku č.117) , model se může zcela zdeformovat a je potřeba manuálně se ho pokusit opravit, v tomto případě je podle mě už většinou jednodušší udělat animaci celou ručně nebo se vrátit do iPi Studia a prostě animaci upravit přímo na nový rig. Druhá možnost je, že na import animace model prostě nebude reagovat vůbec jako model Scouta z Team Fortress 2 na obrázku.



Obrázek 118: Manuální retargeting s využitím constraints

Druhá možnost je animaci namapovat na správný model a následně ji pomocí omezení (constraints) postupně přemapovat na jiný model, tato možnost je sice funkční, ale opět zbytečně zdlouhavá a nepraktická. V modelu s animací vyberu jednu z kostí a v druhém modelu najdu korespondující kost, kterou také vyberu přidržením ctrl a levého tlačítka myši, kliknu na vybranou kost v modelu, na který chci animaci dostat pravým tlačítkem myši a vyberu DAG Utilities menu - constraint orient (někdy také constraint point), nyní by se tato část modelu měla pohybovat stejně jako na původním modelu s animací a v menu s výběrem kostí se objeví tyto nově vytvořená omezení. Takto postupuji, dokud neudělám omezení na všech možných korespondujících kostech, problém je, že s takto uzamčenými kostmi se nedá manuálně vůbec manipulovat a musí být uzamčeny v pozici a následně omezení odstranit z modelu, tím se zachová aktuální pohyb a bude možné s modelem opět normálně pracovat. Tento způsob dosáhl mnohem lepších výsledků než předchozí (viz. obrázek č.118 a obrázek č.119).



Obrázek 119: Přímý import animace vs. manuální retargeting

7.6 Export z iPi studia do Blenderu

V případě exportu do Blenderu je více možností jak postupovat, protože Blender dokáže zpracovat mnohem větší spektrum formátů. Nabízí se export především ve formátu BVH, který se používá primárně přímo pro Mo-cap data a FBX. Je dokonce možné rovnou exportovat data z defaultního iPi rigu a zpracovat je až v Blenderu při mapování na 3D model, nebo samozřejmě z Blenderu exportovat FBX s 3D modelem a ten importovat do iPi a namapovat ho v iPi studiu.

Vyzkoušel jsem všechny možnosti a všechny fungovaly dobře, mojí preferovanou možností bylo namapovat si v iPi studiu FBX model a ten pak opět importovat do Blenderu, vzhledem k tomu, že sekvence jsem mohl potom rovnou renderovat bez dalších úprav.

7.7 Zpracování sekvencí v Blenderu

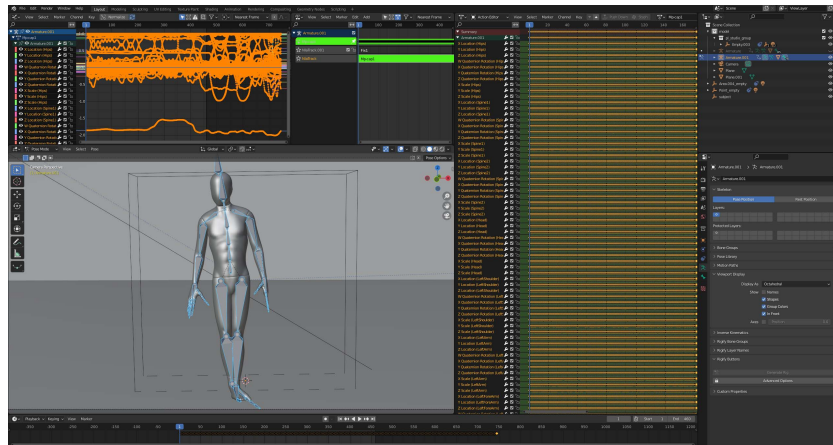
Na rozdíl od SFM, v Blenderu je několik způsobů, jak získanou animaci dostat na model Mannequin. Vzhledem k možnostem Blenderu, kde je možné každý jednotlivý komponent modelu a rigu rozložit na základní prvky a vše změnit, je uživatel limitován v podstatě jen vlastním porozuměním principů, jak jednotlivé systémy Blenderu pracují a svou fantazií.

Je například možné importovat BHV nebo FBX z původního rigu odebrat vliv modelu a připnout ho na rig nový, pokud jsou skupiny vertexů stejné a stejně pojmenované části rigu, model se sám namapuje na nový rig, je možné manuálně části rigu nebo i skupiny vertexů přejmenovat a upravit, aby tento způsob fungoval, je možné uzamknout jednotlivé kosti původního rigu na nový a tím zkopírovat jeho pohyb, je možné v Dope Sheet editoru akci na Mocap rigu přepnout na původní a pokud jsou stejné, pohyb tím přesunout na zvolený model, je možné využít placený addon Auto-Rig Pro, který má jednu svou část přímo zaměřenou na retargeting animací a velmi ho usnadňuje i mezi rozdílnými rigy a existují i mnohé další možnosti, jak dosáhnout v Blenderu svého cíle.

V tomto případě mi přišlo nejjednodušší prostě v iPi namapovat na FBX animaci a do Blenderu importovat.

7.8 Možnosti úprav importované sekvence

Možnosti úpravy importovaných sekvencí jsou v podstatě neomezené, Blender umí jednoduše uložit různé pohyby jako samostatné akce a ty na sebe vrstvit a propojovat je. V nelineárním editoru na sebe je možné pro každý objekt vrstvit na sebe animace a každou z nich kdykoliv upravit.

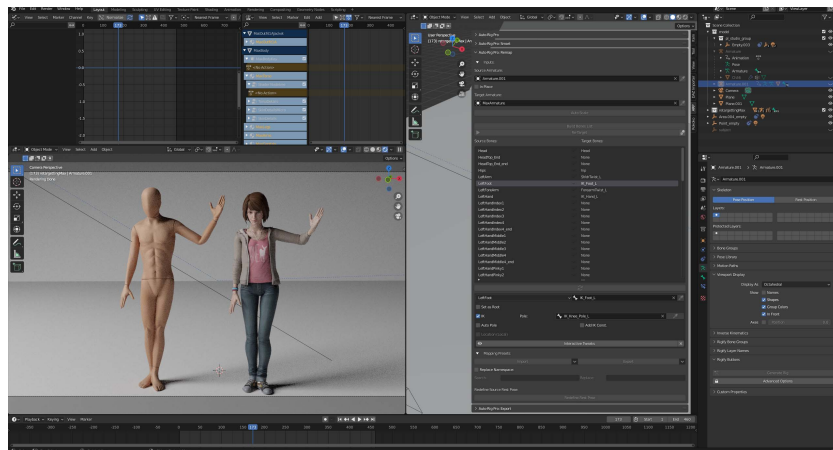


Obrázek 120: Úprava sekvencí v Blenderu

Importované sekvence mají pro každou jednotlivou část rigu v každém snímku vytvoření klíčový snímek na všech osách pro pozici, rotaci i velikost (viz. obrázek č.120) a všechny se dají v grafickém editoru nebo dope sheet jednoduše upravovat a tím se zbavit chyb v originálním snímání.

7.9 Retargeting animace

Opět existuje několik možných přístupů, kterými lze v různých situacích dosáhnout požadovaného výsledku. Pokud je rig modelu podobný, je možné jednoduše na kosti nového rigu zkopírovat snímky z jednotlivých vrstev nelineárního editoru ve stejném pořadí a dostat animaci na nový rig, většinou byl přenos přesný, pokud se jednalo o podobné modely. Pokud jsou modely více rozdílné a rigy také, dobře se mi osvědčil placený addon Auto Rig Pro, který umožňuje poměrně jednoduše namapovat části různých rigů a počítá i s přenosem mezi FK a IK (Inverse Kinematics), jak je možné vidět na obrázku č.121.



Obrázek 121: Retargeting na různé rigy pomocí addonu Auto rig Pro

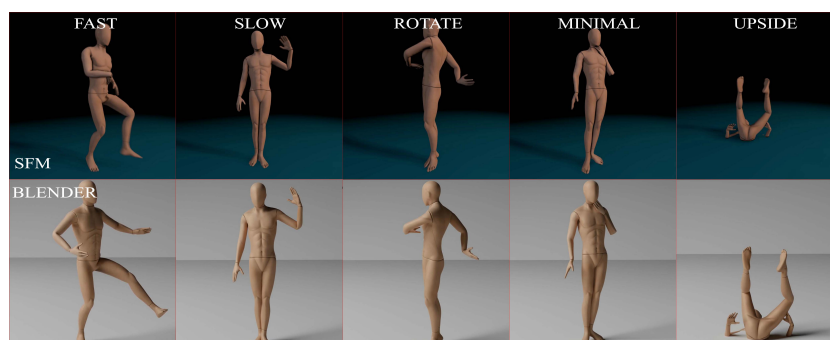
7.10 Blender vs. SFM

Mým záměrem bylo porovnat, jaký je rozdíl ve zpracování importovaných sekvencí programem, jaké rozdíly jsou v možnostech úprav těchto sekvencí a jak snadné nebo složité je aplikovat animaci na jiný než původní model s rozdílným rigem.

7.10.1 Zpracování surových sekvencí

Pro účely porovnání zpracování surových sekvencí jsem v SFM na model Mannequin importoval všechny testovací sekvence jako předtím v Blenderu a vyexportoval je se stejnými parametry, všechny sekvence v rozlišení 1920 x 1080 při 30 snímcích za vteřinu, což je shodná rychlost jako při snímání, abych nenarušil plynulost pohybu.

V SFM jsem také připravil podobnou jednoduchou scénu jako v Blenderu, jednodušší plocha jako podlaha a jednoduché dvoubodové osvětlení modelu. Na první pohled je patrný rozdíl kvality v renderování, protože SFM používá již velmi zastaralý renderovací engine, kdežto Blender využívá moderní technologie s renderovacím enginem Cycles, který jsem pro renderování zvolil. Mě ovšem zajímali spíše rozdíly ve zpracování samotného pohybu a rozdíly v deformaci 3D modelu než grafické porovnání kvality.



Obrázek 122: Zpracování surových sekvencí SFM vs. Blender

Po vyexportování všech dvaceti sekvencí a jejich důkladném porovnání mezi sekvencemi vyexportovanými z Blenderu jsem dospěl k závěru, že oba programy data ze surových sekvencí zpracovávají naprosto stejně, pohyb vypadal úplně stejně bez odchylek mezi oběma obrazy. Protože byl model lehce jinak natočený a kamera v SFM měla nastavený o něco větší úhel, který zabírala, mohlo se jevit že jsou v něčem rozdílné, ale to byl jen vizuální klam.

Mohu říci, že je úplně jedno, do kterého z těchto dvou programů animační sekvenci importuji, výsledný pohyb bude totožný a stejně kvalitní.

7.10.2 Výhody a nevýhody zpracování v Blenderu a SFM

Kde jsou rozdíly naprosto zřejmé je ovšem možnost dalšího zpracování sekvencí po importu, v SFM jsou extrémně omezené a nepraktické, kdežto v Blenderu jsou možnosti dalších úprav sekvencí v podstatě neomezené a jednoduše přístupné, je tedy jednoduché sekvence manuálně vyčistit od chyb, pokud na to uživatel má potřebné znalosti.

Ve všech ohledech je Blender kvalitnější program s většími možnostmi a SFM je již poměrně zastaralý a dlouhé roky neaktualizovaný software, který se ovšem stále těší obrovské popularitě mezi začínajícími animátory.

Můj názor po vyzkoušení práce s Mo-cap sekvencemi s oběma programy je, že pro SFM se hodí pouze, pokud jsou sekvence již téměř perfektně zpracované, protože jejich úprava je v SFM prostě velmi omezená, je možné sekvenci naimportovat do Blenderu, tam ji vyčistit a poté ji exportovat jako DMX pro SFM, ale v tom případě uživatel rovnou může sekvenci vyexportovat přímo z Blenderu jako video

soubor nebo sekvenci PNG obrázků a SFM zcela vynechat.

7.10.3 Motion Capture vs. ruční animace

Po všech těchto zpracovaných sekvencích mám podle mě docela jasný obrázek o tom, jak vhodné je využít domácí Mo-cap s Kinecty a kdy využít manuální animaci. Při porovnání vycházím ze čtrnácti let zkušeností, po které se manuální animaci věnuji.

Manuální animace je zdlouhavá oproti Mo-cap, kdy je možné získat kvalitní základ animované sekvence v reálném čase, zatímco herec provádí pohyb, plus čas, který zabere zpracování těchto dat. Jako příklad mohu uvést například tvorbu cyklu chůze, tento cyklus se bude neustále opakovat a je tedy potřeba ho začít a skončit identickým snímkem mínus jeden snímek, aby se nezduplicoval pohyb na dvou snímcích. Tento cyklus mě osobně po mnoha letech práce s manuální animací může trvat minimálně okolo třiceti minut, abych měl použitelný výsledek. U Mo-cap s Kinecty, když nebudu počítat čas přípravy studia, zabere nasnímání a prvotní zpracování tohoto cyklu i s exportem do Blenderu okolo pěti minut.

U každé Mo-cap sekvence je ale potřeba stejně provést manuální úpravu, nepovedlo se mi nasnímat ani jednu, která by byla hned použitelná bez chyb. Vyčistit tyto chyby a správně tuto animaci zacyklit může zabrat nakonec i více času než samotná manuální animace.

Výsledek zpracované Mo-cap sekvence mi ale vždy přišel mnohem realističtější než při použití čistě manuální animace. Je to kvůli těm drobným detailům, nad kterými se herec ani nezamyslí, kdy nevědomě lehce pootočí částí páteře, lehce trhne rukou a spoustu dalších minimalistických pohybů, které při manuální animaci sice je možné pokusit napodobit, ale i tak nikdy nebudou nejspíš působit zcela přesvědčivě.

Také je třeba vzít v úvahu, že manuální animace uživatele nic nestojí, pro Mo-cap je potřeba vlastnit nějaké reálné vybavení a čím chce uživatel přesnější výsledky, může se rychle dostat do řádů přinejmenším desetitisíců korun, s profesionálními řešeními Mo-cap ještě mnohem výš.

Dalším faktorem je náročnost manuální animace, většina animátorů zdokonaňuje svou techniku i několik let, než výsledky jejich snažení začnou být působivé

a použitelné pro komerční projekty. Udělat kvalitní manuální animaci vyžaduje ochotu se ji naučit a velkou dávku trpělivosti a je potřeba skutečně porozumět pohybu, aby bylo možné ho přesvědčivě napodobit.

U Mo-cap s Kinecty se může v podstatě každý dobrat zajímavých výsledků i bez předchozích zkušeností s animací a je nutné se naučit jen základ pro zpracování těchto sekvencí. Ovšem pro jejich úpravu je již stejně potřebná určitá dávka znalostí a zkušeností s manuální animací, aby se uživatel dostal ke skutečně použitelným výsledkům.

Ruční animace	Motion Capture
- zdlouhavá	- okamžité výsledky
- absolutní kontrola	- nutnost manuálních úprav
- zdarma	- finanční náročnost
- nepřirozenost pohybu	- velmi realistický pohyb
- velmi náročná	- v základu nepříliš náročná

Obrázek 123: Zjednodušené porovnání manuální animace a Mo-cap

V případě jednoduchých přirozených pohybů se podle mě Mo-cap v domácím prostředí s Kinecty osvědčilo dobře a může dosáhnout kvalitních výsledků, které se po menší manuální korekci dají použít pro počítačové hry nebo amatérské animované snímky, a tyto sekvence budou působit velmi přirozeně, pravděpodobně mnohem více, než by většina animátorů dokázala udělat manuálně i za mnohem delší dobu než zabere samotné snímání. V tomto případě si myslím, že je Mo-cap s Kinecty jednoduše skvělý nástroj a je vhodné ho použít.

Je možné s ním udělat i jednoduchou interakci s objekty případně i s více herci, pokud uživatel bude počítat s rozsáhlejší a delší manuální korekcí a dobrat se podobně dobrých použitelných výsledků. To platí pouze pokud mluvím o mapování na humanoidní bipedální postavy.

V případě rychlých hektických pohybů, potřeby pohybu postavy po velkém prostoru, nutnosti interakce několika postav, animace zvířete nebo jiného než humanoidního objektu je použití Mo-cap s Kinecty podle mě nevhodné a výsledky z něj jsou pro tyto účely prakticky nepoužitelné. V tomto případě shledávám jako jednodušší možnost pohyb animovat kompletně ručně jako lepší. Pro Mo-cap rych-

lého pohybu v domácím prostředí bych možná doporučil místo Kinectů PS Eye senzory, které mají vyšší FPS a dohlédnou dál, takže je možné snímat na větší ploše. Ovšem PS Eye už nedisponují Depth kamerou a je nutné pracovat pouze s RGB daty.

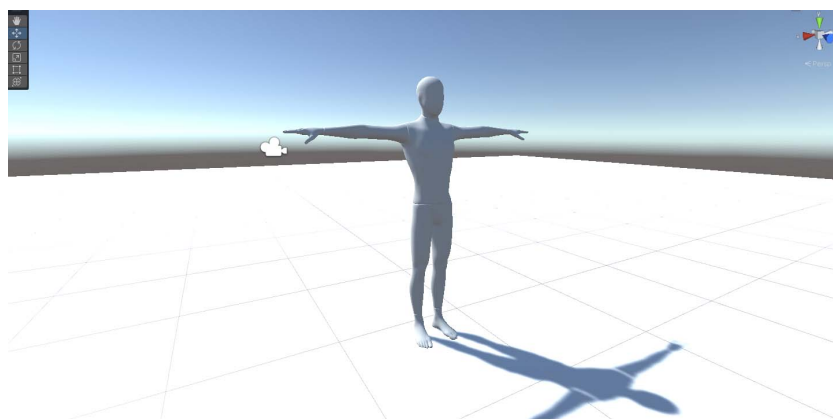
U většiny sekvencí je nutná nějaká forma manuálního zásahu, takže kombinace Mo-cap a ruční animace se zdá jako možnost, jak dosáhnout ve většině případů těch nejlepších výsledků v případě animace humanoidního charakteru.

8 Tvorba aplikace v Unity 3D

V této sekci jsem se chtěl zaměřit na dvě základní věci, import připravených animací z mého Mo-cap do Unity a jejich možné využití a taky na vývoj aplikace, která by využívala přímo Kinect jako ovládací prvek.

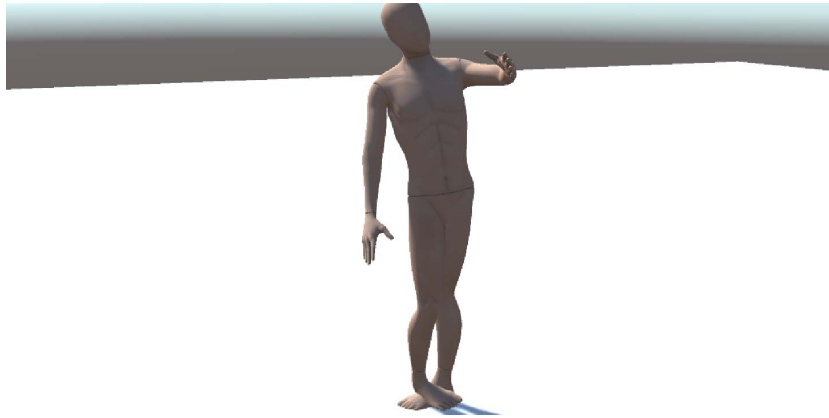
8.1 Import sekvencí do Unity Engine

Vytvořil jsem nový Unity projekt, jako první jsem do Unity potřeboval dostat model Mannequin, ten je přímo uzpůsobený pro Unity, možné stáhnout ze stránek mixamo, takže jsem se rozhodl této možnosti využít. Model jsem vložil do složky Character, kterou jsem si vytvořil ve složce projektu. V Unity jsem si ve složce Assets také vytvořil složku Characters a do ní importoval model Mannequin, na scéně jsem vytvořil plochu reprezentující podlahu a přetáhl na ní model Mannequin, pro podlahu jsem vytvořil nový materiál a aplikoval ho na ní.



Obrázek 124: Unity - importovaný model s vadným materiálem

vstupního stavu.



Obrázek 127: Unity - zpracovaná nasnímaná sekvence přehrávající se na modelu

Na model ve scéně jsem připojil tento nově vytvořený ovladač a po spuštění se na modelu úspěšně přehrávala moje nasnímaná animace (viz. obrázek č.127). Pro zkoušku jsem stejným způsobem aplikoval ještě několik sekvencí na další kopie modelu a všechny fungovaly správně. V tuto chvíli je tedy možné všechny sekvence využívat jako klasické animace, pracovat s nimi v Unity a použít je při vývoji videoher.

8.2 Vývoj aplikace v Unity

Když jsem dostal do Unity nasnímané sekvence, původně jsem chtěl pracovat s nimi na vytvoření malé aplikace. Poté, co jsem je do Unity úspěšně importoval, a použil pro animaci postavy, jsem se nad tímto postupem zamyslel a uvědomil si, že to by nedávalo příliš velký smysl. Důvod je jednoduchý, v tu chvíli už nebyl žádný rozdíl mezi používáním těch nasnímaných sekvencí a klasických předpřipravených animací, nevypovídalo by to již zhora nic o použití Kinectů, pouze bych ukazoval, jak se v Unity pracuje s normálními animačními sekvencemi.

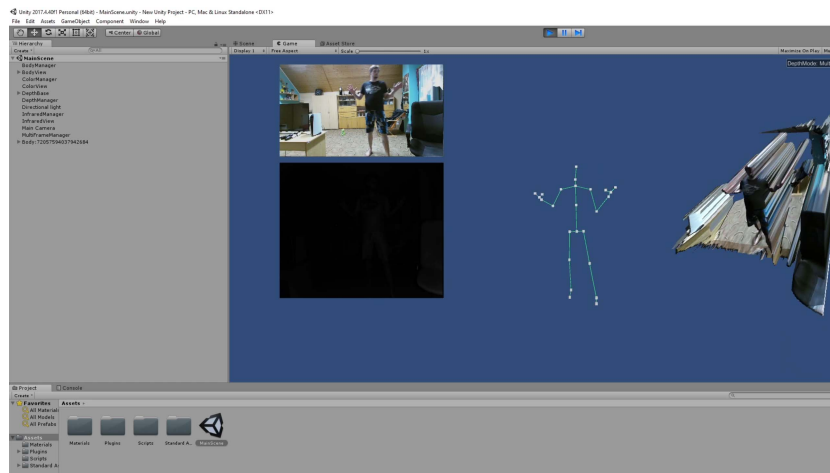
Rozhodl jsem se tedy pro něco úplně jiného a podle mě mnohem zajímavějšího. Moje otázka byla, jestli je možné využít Kinect jako ovladač pro aplikaci vytvořenou v Unity, něco jako byly staré hry pro Xbox.

Při svém průzkumu jsem narazil na nástroje pro Unity, které měly umožňovat vývoj aplikací s pomocí Kinectů v2, jednalo se o balíčky Unity Pro dostupné ze

stránek Microsoftu. Po stažení těchto balíčků a jejich rozbalení jsem narazil hned na několik problémů, nejhorší byla nedostupná dokumentace a fakt, že nástroje jsou zastaralé a neaktualizované.

U balíčků byl jen krátký textový soubor popisující proces, jak zprovoznit nástroje s Unity, tento proces byl napsaný pro několik let starou verzi Unity a již se nedal použít. Hledáním střípků informací po internetu a metodou pokus omyl jsem si udělal představu o tom, jak nástroje zprovoznit, a co který z poskytnutých scriptů umožňuje. Také jsem zjistil, že vzhledem k rozsáhlým změnám v Unity v posledních několika letech, nástroje nejsou s novými verzemi funkční a ve scriptech se objevují stovky chyb. To jsem samozřejmě řešit nechtěl, pokusil jsem se tedy stáhnout starší verzi Unity z roku 2017, konkrétně verzi 2017.4.40f1.

V této verzi jsem vytvořil nový 3D projekt nazvaný Coin Catcher - Kinect v2 a do jeho Assetů vložil balíček Kinect.2.0.1410.19000 a potvrdil import. Poté jsem z balíčků Unity Pro importoval ještě všechny soubory ze složky KinectView, které umožňují práci s hloubkovými a RGB daty Kinectu, tracking postav a obsahují scripty, které budu pro tvorbu své aplikace potřebovat.



Obrázek 128: Unity - KinectView funkční předpřipravená scéna

Po spuštění předpřipravené scény jsem spustil hru a přesvědčil se, jestli spojení s Kinectem funguje. Spojení s Kinectem nefungovalo, uvědomil jsem si, že jsem nahradil původní Microsoft ovladač libfreenect2, a s tím zřejmě Unity neumí pracovat, takže jsem tuto změnu musel vrátit. Po odinstalování libfreenect2 a opět

tovné instalaci oficiálního ovladače se v herním okně objevila postava trackovaná Kinectem, spolu s výstupem z RGB kamery a dalších (viz. obrázek č.128). Pro mé potřeby bude potřeba pouze script BodyView a BodyManager, ostatní jsem ze scény proto odstranil.

Aplikace měla být jednoduchou hrou, ve které hráč pomocí pohybu svých rukou sbírá mince, které se objevují na obrazovce, a tím se mu zvyšuje skóre, pro mé účely v rámci této práce by takový prototyp měl být víc než dostatečný. Veškeré scripty jsou napsané v C#, takže i ostatní potřebné scripty napíšu v tomto programovacím jazyce.

Protože jsem k nástrojům neměl žádnou dokumentaci, hledal jsem opět jakékoliv informace a tutoriály na internetu, abych si udělal přehled o možnostech nástrojů, a jak správně psát script s tímto rozšířením Unity. Nejvíce mi pomohly tutoriály z youtube kanálů VR with Andrew a Code to Create, s jejichž pomocí se mi povedlo úspěšně modifikovat kód a aplikaci zprovoznit.

```
1 using UnityEngine;
2 using UnityEngine.Networking;
3 using System.Collections.Generic;
4
5 using Windows.Kinect;
6 using Joint = Windows.Kinect.Joint;
7
8 Skript Unity | Počet odkazů: 0
9 public class BodySourceView : MonoBehaviour
10 {
11     public BodySourceManager mBodySourceManager;
12     public GameObject mJointObject;
13
14     private Dictionary<ulong, GameObject> mBodies = new Dictionary<ulong, GameObject>();
15     private List<JointType> _joints = new List<JointType>
16     {
17         JointType.HandLeft,
18         JointType.HandRight,
19     };
20
21     Zpráva Unity | Počet odkazů: 0
22     void Update()
23     {
24         #region Get Kinect data
25         Body[] data = mBodySourceManager.GetData();
26         if (data == null)
27             return;
28
29         List<ulong> trackedIds = new List<ulong>();
30         foreach (var body in data)
31         {
32             if (body == null)
33                 continue;
34
35             if (body.IsTracked)
36                 trackedIds.Add(body.TrackingId);
37         }
38     }
39     #endregion
40 }
```

Obrázek 129: Coin Catcher - ukázka scriptu zpracovávajícího data z BodyManager

Script `BodySourceView` zpracovává data z `BodySourceManager` a stará se o zobrazení výstup, tento script jsem tedy potřeboval modifikovat, aby místo kostry celého těla zobrazoval a trackoval pouze ruce hráče a aby je na herní obrazovce vykresloval jako zvolené sprity. Nakreslil jsem si sprity představující ruku a importoval ji do Unity, script jsem upravil tak, aby vyhledával a vykresloval pouze pozici rukou a nastavil jim nové chování, kdy se má sprite rukou na obrazovce objevit a kdy se zničit.

Po spuštění aplikace se na obrazovce nyní vykreslovaly mé sprity rukou, které bylo možné pohybem vlastních rukou ovládat a zanikaly, pokud nebyl ničí pohyb trackován.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  Skript Unity | Počet odkazů: 2
7  public class Score : MonoBehaviour {
8      public static int scoreAmount;
9      private Text scoreText;
10     Zpráva Unity | Počet odkazů: 0
11     void Start () {
12         scoreText = GetComponent<Text>();
13         scoreAmount = 0;
14     }
15     Zpráva Unity | Počet odkazů: 0
16     void Update () {
17         scoreText.text = "Score: " + scoreAmount;
18     }
19 }
```

Obrázek 130: Coin Catcher - ukázka scriptu zajišťujícího počítání hráčova skóre

Dalším krokem bylo vytvoření scriptu pro generaci peněz na obrazovce, pro ně jsem vytvořil nový script `CoinBeh` a `CoinGen`, ve kterých jsem nastavil generování mincí, interval vytvoření nových mincí, jejich pohyb, rozmezí pozic, kde se mohou generovat a podmínku zmizení mince z obrazovky. Vytvořil jsem pomocné scripty `Hand` a `Score`, ve kterých jsem se postaral o nahrazení sprity mince za sprite sebrané mince a přičtení mince k celkovému skóre, které jsem zobrazoval na UI

Text ve scéně.

Ke hře jsem vytvořil obrázek pozadí, importoval ho do Unity a nastavil ho, aby se zobrazoval před hlavní kamerou, tím byl prototyp hry hotový, takže jsem vyexportoval aktuální build aplikace.

8.3 Představení vytvořené aplikace

Prototyp Coin Catcher funguje tak, jak jsem chtěl, využívá Kinect v2 jako ovladač, dokáže trackovat ruce i více osob najednou a všichni můžou sbírat mince, které se tvoří na obrazovce, přičemž každá sebraná mince zvyšuje skóre hráčů. Vytvořený prototyp aplikace se nachází v příloze A na přiloženém DVD.



Obrázek 131: Coin Catcher - ukázka prototypu

Na tomto prototypu by se dalo stavět a vytvořit zábavnou menší hru, stačilo by přidat hlavní menu, přidat mód pro víc hráčů, kdy by hra počítala skóre každému hráči zvlášť, případně kromě mincí nějaké jiné položky, kterým by se hráč naopak musel vyhýbat, více typů mincí, závěrečnou obrazovku, žebříček se skóre, omezit herní čas, aby hráč musel sebrat co nejvíce mincí v limitu, přidat speciální položku, co by umožnila hráči při sebrání herní čas o něco navýšit, a dalo by se vymyslet spoustu dalších věcí, jak hru vylepšit. Ty už ovšem nesouvisí s využitím Kinectu a tento prototyp jasně dokazuje, že je možné s ním v rámci Unity pracovat.

Je zde několik nepříjemných překážek, od zastaralých a neudržovaných nástrojů, chybějící dokumentace a faktu, že by si takové aplikace našly velmi málo

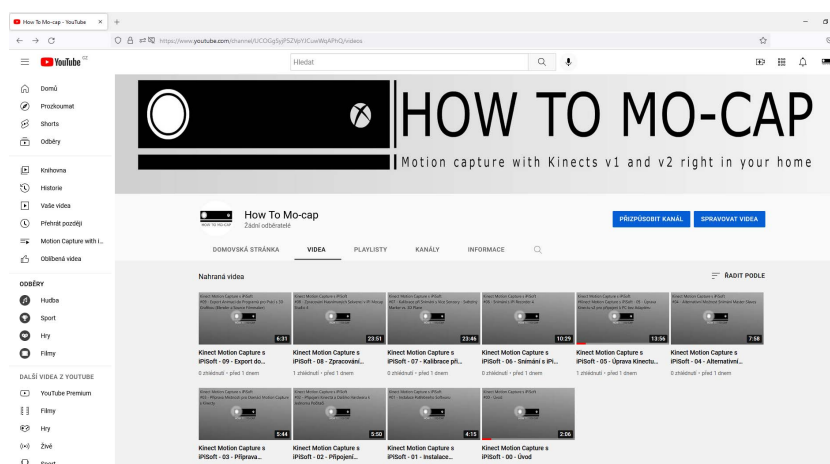
hráčů, protože konzolový Kinect se nedá bez adaptéru připojit k jiným zařízením než Xboxům a nejsem si jistý, kolik by se našlo hráčů, co ho běžně využívají jako periférii pro hraní na PC.

9 Tvorba tutoriálů a webu

9.1 Tvorba tutorial videí

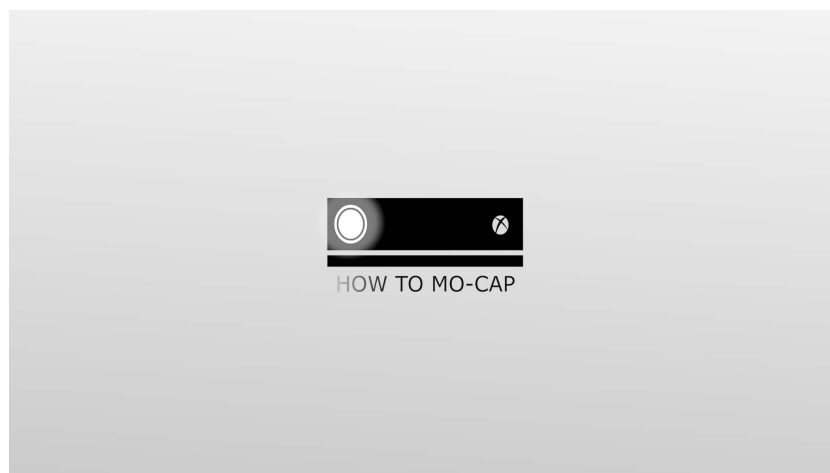
Ze získaných zkušeností jsem se rozhodl vytvořit sérii video tutoriálů věnujících se možnostem domácího Mo-cap. Nejdřív jsem si rozvrhl o jakých tématech chci mluvit a rozdělil si je do jednotlivých videí. Rozhodl jsem se pro sérii deseti videí probírajících postupně postavení domácího studia od úplných základů až po export zpracovaných sekvencí. Z této série jsem se rozhodl vynechat věci jako porovnávání Kinectů a počtu senzorů mezi sebou nebo snímání speciálních typů sekvencí (ty jsem se rozhodl udělat v samostatných videích mimo hlavní sérii tutoriálů) a zaměřil jsem se hlavně na proces nahrávání a zpracování nasnímaných sekvencí. V hlavní sérii se nachází těchto deset videí:

1. 1. Úvod
2. 2. Instalace potřebného softwaru
3. 3. Připojení Kinectů a dalšího hardwaru k jednomu počítači
4. 4. Příprava místnosti pro domácí Motion Capture s Kinecty
5. 5. Alternativní možnost snímání Master-Slave mód
6. 6. Hardwarová úprava Kinectu v2 pro připojení k PC bez adaptéru
7. 7. Snímání s iPi Recorder 4
8. 8. Kalibrace při snímání s více senzory - Světelný marker vs. 3D Plane
9. 9. Zpracování nasnímaných sekvencí v iPi Mocap Studio 4
10. 10. Export animací do programů pro práci s 3D grafikou (Blender a Source Filmmaker)



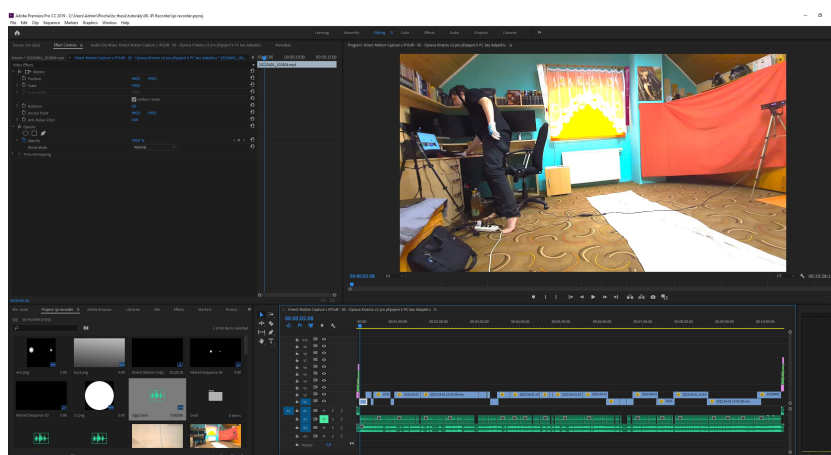
Obrázek 132: Youtube kanál How To Mo-cap

Jako první jsem se rozhodl vytvořit jednotný design pro youtube kanál How To Mo-cap a videa, která na tento kanál umístím. Rozhodl jsem se logo pojmout jako zjednodušenou siluetu Kinectu v2, grafiku jsem vytvořil v Adobe Flash Professional CC a pro videa s nimi naanimoval jednoduché intro v Adobe Premiere Pro a vytvořil audio v Adobe Audition.



Obrázek 133: Intro videí How To Mo-cap

Po vytvoření grafiky jsem si připravil template video projektu, který jsem chtěl použít pro všechny videa v sérii, a začal jsem s prací na úvodním videu.



Obrázek 134: Editování videa v Premiere Pro

Proces tvorby videí se skládal z nahrávání obrazovky počítače při práci v kombinaci se záběry z kamer Kinectů a telefonu při snímání sekvencí. Pro některá videa jsem si po nahrání materiálů napsal script a podle něj stříhal materiál, ke kterému jsem postupně daboval komentář. V některých případech, kdy už byly potřeba pouze záběry z plochy počítače, se mi lépe osvědčilo v OBS nastavit rovnou nahrávání mikrofonu spolu se záznamem obrazovky a bez připraveného scénáře komentovat, co se na ploše počítače momentálně děje.

Po dokončení prací na hlavní sérii videotutoriálů jsem měl hotových 10 videí o celkové délce okolo 110 minut. Tyto videa jsem umístil na youtube kanál How To Mo-cap.

Později jsem vytvořil ještě dvě další videa, která se věnují snímání speciálních sekvencí a porovnání kvality snímání mezi různými konfiguracemi. Tyto videa mají dohromady dalších necelých 50 minut.

9.2 Vytvoření webu *howtomocap.com*

Pro svůj web jsem si registroval doménu *howtomocap.com* u společnosti OneBit Solutions. Nejdřív jsem si rozvrhl, jak by měl web graficky vypadat a jaké by měl mít rozložení a ovládací prvky.

Po vymyšlení hrubého návrhu jsem nejdřív vytvořil potřebné grafické prvky. Jednalo se převážně o rozšíření grafiky z designu youtube kanálu How To Mo-cap. Poté jsem v PSPadu začal psát základní kód webu v html a na něj připojil css.

```

<!DOCTYPE html>

<html lang="cs">

  <head>
    <link rel="stylesheet" href="css/main.css"/>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <meta name="description" content="How to Mo-cap">
    <meta name="keywords" content="Mo-cap, mocap, Motion Capture, Kinect, v1, v2, IPIsoft, Mocap Studio">
    <meta name="author" content="all: Jan Vaněk; janvanek8@seznam.cz"/>
    <meta name="copyright" content="Jan Vaněk (janvanek8@seznam.cz) 2022"/>
    <meta name="robots" content="index, follow">
    <meta name="googlebot" content="index, follow, archive, snippet"/>
    <meta name="language" content="cs-cz"/>
    <link rel="shortcut icon" type="image/x-icon" href="pictures/icon.png" />
    <title>How to Mo-cap</title>
    <link rel="stylesheet" type="text/css" href="css/style_print.css" media="print" />
    <link rel="stylesheet" type="text/css" media="screen and (min-width: 2100px)" href="css/style_big.css" />
    <link rel="stylesheet" type="text/css" media="screen and (min-width: 1024px) and (max-width: 2100px)" href="css/style_screen.css" />
    <link rel="stylesheet" type="text/css" media="screen and (max-width: 1024px)" href="css/style_mobile.css" />
  </head>

  <body>
    <div id="main">
      <header>
        

        <ul id="mainMenu">
          <li id="active"><a href="index.html" title="domů">domů</a></li>
          <li><a href="how_mocap.html" title="domčí Mo-cap">domčí Mo-cap</a></li>
          <li><a href="kinect_comparison.html" title="Porovnání Kinectů">Porovnání Kinectů</a></li>
          <li><a href="special.html" title="Speciální">Speciální</a></li>
          <li><a href="poradna.php" title="Poradna">Poradna</a></li>
        </ul>
      </header>

      <main>
        <nav>
          <ul id="sideMenu">
            <li><a href="novinky.html" title="novinky">Novinky</a></li>
            <li><a href="faq.html" title="faq">F.A.Q.</a></li>
            <li><a href="kontakt.html" title="kontakt">kontakt</a></li>
          </ul>
        </nav>

        <aside>
          <ul id="doprva">
            <li id="special">
              <li><a title="howtomocap"><a href="https://www.youtube.com/channel/UC0G5yJP52Vp7CuuuqPHQ" * target="_blank" rel="noopener noreferrer" title="howtomocap"></a></li>
              <li><a title="ipicture"><a href="https://www.ipi-soft.com/store/" * target="_blank" rel="noopener noreferrer" title="IPI Store"></a></li>
              <li><a title="ipiwiki"><a href="https://wiki.ipi-soft.com/Main_Page" * target="_blank" rel="noopener noreferrer" title="IPI wiki"></a></li>
            </ul>
          </ul>
        </aside>
      </main>
    </div>
  </body>
</html>

```

Obrázek 135: Ukázka HTML kódu hlavní stránky

Vytvořil jsem tři css soubory jeden, pro mobilní zařízení, jeden pro počítače a jeden pro neobvykle velké obrazovky. Nejdřív jsem vytvořil hlavní stránku webu a vychytil rozložení a odkazy všech menu. Poté jsem vytvořil stránku pro každou položku hlavního menu a následně postupně rozšiřoval každou z těchto hlavních stránek do podstránky.



```
styly_screen.css - [C:\Users\janva\Desktop\howtomocap web\css\styly_screen.css]
Soubor Projekt Úpravy Hledat Zobrazit Formát Kódová stránka Nástroje HTML Nastavení Okno
1.. index.html 2.. styly_screen.css
Nový proje
  Složka
ul#mainMenu {
padding:0;
list-style-type:none;
text-align: center;
width:100%;
background-color: rgba(170,170,170,0.7);
height:50px;
border-bottom: solid 5px #fff;
font-size:20px;
margin-top: -5px;
}

ul#mainMenu li {
list-style-type:none;
width:20%;
float:left;
margin-right: 28px;
text-align:center;
border-left: solid 1px #fff;
border-right: solid 1px #fff;
}

ul#mainMenu li a{
text-decoration:none;
color:white;
line-height:50px;
display:block;
}

ul#mainMenu li a:hover{
background-color: #fd0;
color:#000;
-webkit-transition: all 1s ease;
-moz-transition: all 1s ease;
-o-transition: all 1s ease;
-ms-transition: all 1s ease;
transition: all 1s ease;
}

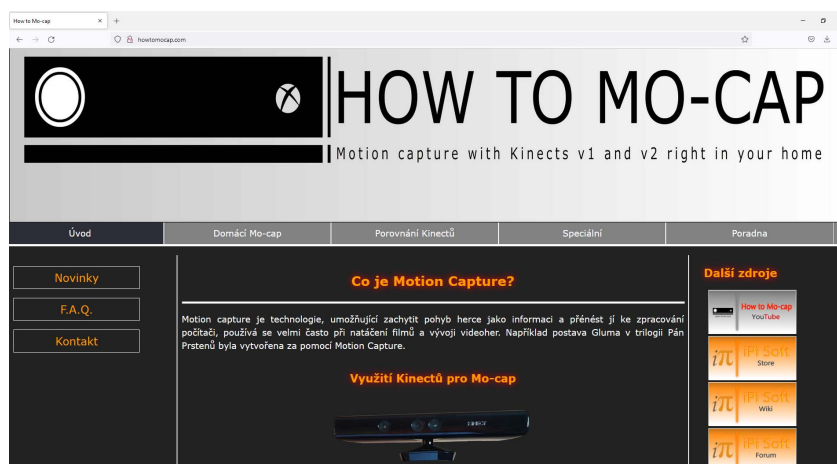
img#logo {
width: 100%;
height: auto;
max-height: 500px;
}

ul#sideMenu {
```

Obrázek 136: Ukázka CSS kódu webu *howtomocap.com*

Na webu se nachází rozdělené do kategorií informace potřebné pro vytvoření domácího Mo-cap studia s Kinecty, mé výsledky z porovnávání různých konfigurací Kinectů, odkaz na sérii videotutoriálů o Mo-cap a další užitečné informace, včetně

poradny pro návštěvníky webu.



Obrázek 137: Hlavní stránka webu *howtomocap.com*

Hotový web jsem poté nahrál na FTP server. Zdrojové kódy vytvořeného webu se nachází v příloze A na přiloženém CD, stejně tak vytvořené videotutoriály.

10 Zhodnocení a závěr

Bakalářská práce se zabývala vytvořením domácího Motion Capture studia, zkoumáním kvality snímání při použití různého počtu Kinect sensorů a mezi Kinecty v1 a v2, úpravou hardwaru konzolové verze Kinectu v2 pro připojení k PC bez oficiálního adaptéru, použitím Mo-cap sekvencí v programech Blender a Source Filmmaker, využitím Kinectů při vývoji aplikací v Unity a přiblížení informací o domácím Mo-cap širšímu publiku.

V teoretické části jsem se zaměřil na vysvětlení principu technologie Motion Capture, na rozdíly mezi jednotlivými druhy Kinectů a rozdíly oproti klasickým RGB kamerám, představení vybraného softwaru a rozdíly mezi profesionálními Mo-cap řešeními a Mo-cap studiem používajícím Kinecty.

V praktické části jsem sehnal potřebný hardware, připravil snímací místnost a zřídil v ní domácí hybridní Mo-cap studio využívající Kinecty jako optické senzory a Joy-cony jako motion-controllery.

Nasnímal jsem pro hlavní porovnávací část celkem 20 testovacích sekvencí 5ti typů pohybu při čtyřech konfiguracích Kinectů, tyto sekvence jsem zpracoval, vyexportoval do Blenderu a vyrenderoval do video souborů a výsledky jsem vyhodnotil. Poté jsem se věnoval ještě několika speciálním typům snímání, pro které jsem také vyhodnotil výsledky stejným způsobem.

Dále jsem se zaměřil na možnosti exportu zpracovaných sekvencí do programů pro práci s 3D grafikou Blender a Source Filmmaker, vyzkoušel namapování sekvence na různé typy rigů a prozkoumal možnosti dalších úprav v Blenderu a SFM a možnosti retargettingu animace na další 3D modely.

Poté jsem ještě prozkoumal možnost využití zpracovaných sekvencí v Unity a možnost použití Kinectu jako ovladače pro aplikaci vytvořenou v Unity s pomocí sady nástrojů pro Unity od Microsoftu. V rámci tohoto pokusu jsem vytvořil prototyp jednoduché aplikace nazvanou Coin Catcher - Kinect v2.

Po dokončení praktické části jsem díky získaným zkušenostem vytvořil sérii video tutoriálů, obsahující 10 videí o celkové délce okolo 110 minut a věnující se vytvoření domácího Mo-cap studia od instalace softwaru až po export zpracovaných sekvencí. Kromě tohoto tutoriálu jsem vytvořil ještě několik dalších samostatných videí, týkajících se porovnávání kvality snímání s různými konfiguracemi Kinectů

a snímání speciálních sekvencí.

Zároveň jsem si zaregistroval doménu howtomocap.com a napsal web, na kterém publikuji své výsledky a závěry, poskytuji informace ohledně domácího Mo-cap a také provozuji poradnu pro uživatele, kteří se o domácí Mo-cap zajímají.

Při své práci jsem se toho naučil skutečně hodně a získal mnoho užitečných zkušeností. Zjistil jsem, že Kinecty jsou vhodné pro snímání určitých typů pohybu, ale nezvládají snímání jakýchkoliv rychlých pohybů kvůli svému omezení na rychlost snímání při 30 snímcích za vteřinu. Zjistil jsem, že při kombinaci Mo-cap s Kinecty a lehké manuální korekce je možné dosáhnout poměrně kvalitních realistických ztvárnění pohybu člověka pro užití ve vývoji Indie videoher nebo filmů. Jako animátorovi se mi získané zkušenosti budou hodit při vývoji mých dalších projektů a doufám že se díky mnou vytvořeným video tutoriálům a webu věnujícímu se Mo-cap povede rozšířit povědomí o možnostech domácího Mo-cap mezi více uživatelů.

Seznam použité literatury a zdrojů

- [1] TÖLGYESSY, Michal, Martin DEKAN, Luboš CHOVANEC a Peter HUBINSKÝ. Evaluation of the Azure Kinect and Its Comparison to Kinect V1 and Kinect V2 [online]. Ilkovičova 3, 812 19 Bratislava, Slovakia, 2021 [cit. 2022-03-08]. Dostupné z: <https://pdfs.semanticscholar.org/be5d/e49ab549f0ca21a1ed43bef079194c2fd55b.pdf>. Nstitute of Robotics and Cybernetics, Faculty of Electrical Engineering and Information Technology STU in Bratislava.
- [2] CONDITT, Jessica. 100 years of motion-capture technology. Endgadget [online]. 25.5.2018 [cit. 2022-03-08]. Dostupné z: <https://www.engadget.com/2018-05-25-motion-capture-history-video-vicon-siren.html>
- [3] STURMAN, David. A Brief History of Motion Capture for Computer Character Animation. Uniovi [online]. [cit. 2022-03-08]. Dostupné z: https://www6.uniovi.es/hypgraph/animation/character_animation/motion_capture/history1.htm
- [4] LUM, Rachel. A Brief History of Motion Tracking Technology and How it is Used Today. Medium [online]. 9.1.2019 [cit. 2022-03-08]. Dostupné z: <https://medium.com/@lumrachele/a-brief-history-of-motion-tracking-technology-and-how-it-is-used-today-44923087ef4c>
- [5] MIKHAILOVSKY, Andrei. 100 years of motion-capture technology. Teslasuit [online]. © 2021 TESLASUIT, 13.5.2017 [cit. 2022-03-08]. Dostupné z: <https://teslasuit.io/blog/motion-capture-what-it-is/>
- [6] JAREŠOVÁ, Miroslava. Zachycení a vizualizace pohybu v reálném čase [online]. Brno, 2012 [cit. 2022-03-09]. Dostupné z: https://is.muni.cz/th/s4700/Jaresova_text_prace.pdf bakalářská práce. MASARYKOVA UNIVERZITA FAKULTA INFORMATIKY.
- [7] SINCLAIR, Brendan. Microsoft discontinues Kinect. Game Industry [online]. 25.10.2017 [cit. 2022-03-09]. Dostupné z: <https://www.gamesindustry.biz/articles/2017-10-25-microsoft-discontinues-kinect>

- [8] BROWN, Matt. Microsoft Kinect returns for Azure, developer kit pre-orders live. Windows Central [online]. 24.2.2019 [cit. 2022-03-09]. Dostupné z: <https://www.windowscentral.com/microsoft-kinect-returns-azure-developer-kit-preorders-live>
- [9] JOSEPH HERMAN, Joseph. Motion Capture for the Masses – iPi Soft Does Away with the MoCap Suit. Cinemontage [online]. 7715 Sunset Boulevard, Suite 200 Hollywood, CA 90046, 15.2.2019 [cit. 2022-03-09]. Dostupné z: <https://cinemontage.org/motion-capture-for-the-masses/>
- [10] Kinect Development Xbox 360 Kinect vs Kinect for Windows. Stackoverflow [online]. 2015 [cit. 2022-03-09]. Dostupné z: <https://stackoverflow.com/questions/28176024/kinect-development-xbox-360-kinect-vs-kinect-for-windows>
- [11] Microsoft Forums [online]. [cit. 2022-03-09]. Dostupné z: <https://social.msdn.microsoft.com/Forums/>
- [12] PROTALINSKI, Emil. Microsoft stops producing Kinect for Windows v2 sensor, will focus on Kinect for Xbox One and Windows apps. Venturebeat [online]. © 2022 VentureBeat, 2.4.2015 [cit. 2022-03-09]. Dostupné z: <https://venturebeat.com/2015/04/02/microsoft-stops-producing-kinect-for-windows-v2-sensor-will-focus-on-kinect-for-xbox-one-and-windows-apps/>
- [13] SMEENK, Roland. Kinect V1 and Kinect V2 fields of view compared. Smeenk [online]. © Roland Smeenk 2022, 11.3.2014 [cit. 2022-03-09]. Dostupné z: <https://smeenk.com/kinect-field-of-view-comparison/>
- [14] iPiSoft [online]. [cit. 2022-03-10]. Dostupné z: <https://www.ipisoft.com>
- [15] iPiSoft [online]. © iPi Soft 2014-2022 [cit. 2022-03-10]. Dostupné z: <https://docs.ipisoft.com/>
- [16] Source Filmmaker. SteamDB [online]. [cit. 2022-03-10]. Dostupné z: <https://steamdb.info/app/1840/graphs/>
- [17] Blender Release Notes. Wiki Blender [online]. [cit. 2022-03-10]. Dostupné z: https://wiki.blender.org/wiki/Reference/Release_Notes
- [18] Unity Real-Time Development Platform | 3D, 2D VR & AR Engine [online]. Unity Real-Time Development Platform | 3D, 2D VR & AR En-

- gine. ©Copyright 2021 Unity Technologies [cit. 2021-8-7]. Dostupné z: <https://unity.com/>
- [19] iPiSoft Forum [online]. [cit. 2022-03-11]. Dostupné z: <https://forum.ipisoft.com>
- [20] Motion Controllers. IPi Docs [online]. (c) iPi Soft 2014-2022 [cit. 2022-03-11]. Dostupné z: https://docs.ipisoft.com/Motion_Controller
- [21] User Guide for Dual Depth Sensor Configuration. iPiSoft Wiki [online]. [cit. 2022-03-11]. Dostupné z: https://wiki.ipisoft.com/User_Guide_for_Dual_Depth_Sensor_Configuration
- [22] User Guide for Multiple Depth Sensors Configuration. iPiSoft Docs [online]. [cit. 2022-03-11]. Dostupné z: https://docs.ipisoft.com/User_Guide_for_Multiple_Depth_Sensors_Configuration
- [23] Kinect for Windows SDK v1.8. Microsoft [online]. © Microsoft 2022 [cit. 2022-03-12]. Dostupné z: <https://www.microsoft.com/en-us/download/details.aspx?id=40278>
- [24] Kinect for Windows SDK 2.0. Microsoft [online]. © Microsoft 2022 [cit. 2022-03-12]. Dostupné z: <https://www.microsoft.com/en-us/download/details.aspx?id=44561>
- [25] Distributed Recording. iPiSoft [online]. (c) iPi Soft 2014-2022 [cit. 2022-03-12]. Dostupné z: https://docs.ipisoft.com/Distributed_Recording
- [26] Libfreenect2. Github [online]. © 2022 GitHub, Inc. [cit. 2022-03-12]. Dostupné z: <https://github.com/OpenKinect/libfreenect2>
- [27] Multiple Kinects v2 on a Single PC. iPiSoft [online]. (c) iPi Soft 2014-2022 [cit. 2022-03-12]. Dostupné z: https://docs.ipisoft.com/Multiple_Kinects_v2_on_a_Single_PC
- [28] SARKAR, Samit. Microsoft discontinues Xbox One Kinect adapter. Polygon [online]. © 2022 Vox Media, LLC., 3.1.2018 [cit. 2022-03-13]. Dostupné z: <https://www.polygon.com/2018/1/2/16842072/xbox-one-kinect-adapter-out-of-stock-production-ended>

- [29] Eagle115. A pictorial guide for converting your Kinect to be compatible with an Xbox One S/X or PC for \$40. #thanksmicrosoft. Imgur [online]. 16.2.2018 [cit. 2022-03-13]. Dostupné z: <https://imgur.com/a/e10dN>
- [30] User Guide for Single Depth Sensor Configuration. iPiSoft [online]. [cit. 2022-03-18]. Dostupné z: https://wiki.ipisoft.com/User_Guide_for_Single_Depth_Sensor_Configuration
- [31] IPi Recorder Setup. iPiSoft [online]. [cit. 2022-03-18]. Dostupné z: https://wiki.ipisoft.com/IPi_Recorder_Setup
- [32] Calibration. iPiSoft [online]. [cit. 2022-03-18]. Dostupné z: <https://wiki.ipisoft.com/Calibration>

Seznam obrázků

1	Kinect senzor v1 (nahore) a v2 (dole)	21
2	Kinect v1 RGB	22
3	Kinect v1 Depth	22
4	Kinect v2 RGB	23
5	Kinect v2 Depth	24
6	Kinect for Windows edice	25
7	Mo-cap studio hardware	31
8	Úprava snímací místnosti - uvolnění snímacího prostoru	36
9	Snímek z hloubkových kamer - nezakryté	38
10	Zakrytí problematických ploch	39
11	Snímek z hloubkových kamer - zakryté	39
12	Označení snímené plochy	40
13	První zapojení Kinectu v1 - chybějící Kinect SDK 1.8	40
14	iPi Recorder 4 - instalace potřebných součástí	41
15	Zapojení Kinectu v1 - nainstalované Kinect SDK 1.8	41
16	iPi Recorder 4 - výběr kamer	42
17	První zapojení Kinect v2 - bez Kinect for Windows SDK 2.0	42
18	Zapojení Kinect v2 - nainstalované Kinect for Windows SDK 2.0	43
19	Kinect v2 - pokus o zapojení více senzorů s oficiálním ovladačem do jednoho PC	43
20	Propojení počítačů pro distribuované snímání	44
21	iPiRecorder 4 - Master-Slaves nastavení	45
22	iPiRecorder 4 - výběr otroků	45
23	Automatické sloučení sekvencí při distribuovaném snímání v iPi Recorderu 4	46
24	Manuální sloučení sekvencí z jednotlivých master a slave iPiVideo souborů	47
25	Typy iPiVideo souborů	47
26	iPi Recorder - více v2 senzorů na jednom PC s využitím libfreenect2	48
27	Joy-con v režimu párování	50
28	Windows 10 - Párování Joy-conu přes Bluetooth	50

29	Joy-con spárovaný s Windows 10	51
30	Porovnání délky kabeláže upraveného Kinectu v2 a oficiálního adaptéru	53
31	Klasické USB-B 3.0 vs. originální Kinect v2 kabel	54
32	Nástroje a komponenty potřebné pro úpravu Kinectu	56
33	Kinect v2 s vytaženým kabelem	57
34	Nahřátí polepů pro snadné odstranění	57
35	Klasický Torx 10 vs. bezpečnostní Torx 10	58
36	Odšroubování bezpečnostního Torx pomocí malého plochého šroubováku	59
37	Rozdílné délky Bezpečnostních Torx 10 (nalevo) a torx 10 (napravo)	60
38	Odebrání dvou skrytých Torx 10 šroubků držících přední a vrchní část krytu Kinectu	61
39	Rozložený Kinect v2	61
40	Hledání pinů vylučovací metodou	62
41	Hledání správných pinů pomocí multimetru	62
42	Zvolené piny pro připojení kabelů	63
43	Cínování	64
44	Pocínované dráty	65
45	Provláčení pocínovaných kabelů za kostřemím	65
46	Připojení kabelů k základní desce Kinectu	66
47	Dráty zapojené do konektoru pro připojení napájecího zdroje	66
48	Kontrola správného propojení konektoru s deskou	67
49	Upravený Kinect v2	67
50	Herec v uplém oblečení kopírující křivky těla	75
51	Uchycení Joy-conu k ruce	76
52	iPi Recorder základní nastavení	77
53	Kalibrace Joy-conů	78
54	iPi Recorder 4 - Evaluace pozadí	79
55	Aplikace evaluovaného pozadí při klíčování	80
56	iPi Recorder 4 - nastavení nahrávání	80
57	Snímání kalibrační sekvence pomocí 3D plochy	82
58	iPi Mocap Studio - okno programu při kalibraci s 3D plochou	83

59	Hlubková data z dvou senzorů před kalibrací	84
60	Úspěšná kalibrace s pomocí 3D plochy	85
61	Upravená svítlna pro použití jako světelná značka	86
62	Kalibrace pomocí světelné značky	87
63	Skutečný a virtuální herec v T-pose	88
64	Kinect senzory v2 a v1 na stativu	90
65	Zarovnání virtuálního herce s hloubkovými daty	93
66	iPi Mocap Studio - nástroje pro úpravu pozice a postavy virtuálního herce	94
67	iPi Mocap Studio - nastavení trackingu	95
68	iPi Mocap Studio - nastavení motion controllerů	97
69	iPi Mocap Studio - vytvoření projektu s více Kinecty	99
70	Model Mannequin z kolekce Mixamo	100
71	Mapování kostí nového Rigu	101
72	Získané sekvence v prostředí Blenderu	102
73	Minimalistický pohyb - jeden Kinect v1	103
74	Minimalistický pohyb - dva Kinecty v1	103
75	Minimalistický pohyb - jeden Kinect v2	104
76	Minimalistický pohyb - dva Kinecty v2	104
77	Normální pohyb - jeden Kinect v1	105
78	Normální pohyb - dva Kinecty v1	106
79	Normální pohyb - jeden Kinect v2	106
80	Normální pohyb - dva Kinecty v2	107
81	Rychlý pohyb - jeden Kinect v1	108
82	Rychlý pohyb - dva Kinecty v1	108
83	Rychlý pohyb - jeden Kinect v2	109
84	Rychlý pohyb - dva Kinecty v2	109
85	Otáčení - jeden Kinect v1	110
86	Otáčení - dva Kinecty v1	110
87	Otáčení - jeden Kinect v2	111
88	Otáčení - dva Kinecty v2	111
89	Vzhůru nohama - jeden Kinect v1	112
90	Vzhůru nohama - dva Kinecty v1	113

91	Vzhůru nohama - jeden Kinect v2	113
92	Vzhůru nohama - dva Kinecty v2	114
93	Velikost herce - rozdílný poměr velikosti hlavy k tělu u dítěte a dospělého	116
94	Snímání malé osoby - zpracovaná sekvence	117
95	Snímání ve tmě - klíčování scény za světla	117
96	Snímání ve tmě - klíčování stejné scény po zhasnutí světel	118
97	Snímání ve tmě - zpracovaná sekvence	118
98	Využití motion controllerů k rotaci hlavy	119
99	Sekvence sezení v židli	120
100	Interakce s malým objektem	121
101	Interakce více herců	122
102	Source Tools Addon	124
103	Kosti rigu přejmenované podle Valve Biped standartu	125
104	Problémy s weight links na vertexech 3D modelu	125
105	Script pro kompilaci modelu Mannequin pro SFM	127
106	Kompilace modelu pomocí Crowbaru	128
107	Crowbar - nastavení cest programu k potřebným komponentům SFM	128
108	Načtení modelu Mannequin do SFM bez materiálů	129
109	Příprava textur pro SFM	130
110	VMT script pro materiál v SFM	131
111	Model Mannequin v Blenderu a SFM	132
112	Export DMX pro práci v iPi Mocap Studiu	133
113	DMX iPi Mocap Studio rig retargeting	134
114	Import animace na model Mannequin v SFM	135
115	SFM Motion Editor	136
116	SFM Graph Editor	136
117	Import animace v SFM na různé modely	137
118	Manuální retargeting s využitím constraints	138
119	Přímý import animace vs. manuální retargeting	139
120	Úprava sekvencí v Blenderu	141
121	Retargeting na různé rigy pomocí addonu Auto rig Pro	142
122	Zpracování surových sekvencí SFM vs. Blender	143

123	Zjednodušené porovnání manuální animace a Mo-cap	145
124	Unity - importovaný model s vadným materiálem	147
125	Unity - importovaný model s opraveným materiálem	148
126	Unity - nový jednoduchý animační ovladač	148
127	Unity - zpracovaná nasnímaná sekvence přehrávající se na modelu .	149
128	Unity - KinectView funkční předpřipravená scéna	150
129	Coin Catcher - ukázka scriptu zpracovávajícího data z BodyManager	151
130	Coin Catcher - ukázka scriptu zajišťujícího počítání hráčova skóre .	152
131	Coin Catcher - ukázka prototypu	153
132	Youtube kanál How To Mo-cap	156
133	Intro videí How To Mo-cap	156
134	Editování videa v Premiere Pro	157
135	Ukázka HTML kódu hlavní stránky	158
136	Ukázka CSS kódu webu howtomocap.com	159
137	Hlavní stránka webu howtomocap.com	160

Seznam tabulek

1	Rozdíly senzorů [1]	20
2	Rozdíly v licencích iPiSoft Mocap Studio 4 [14]	28
3	Cena vytvoření mého domácího Mo-cap studia	33
4	Cena mnou doporučené konfigurace domácího Mo-cap studia	34
5	Cena minimální konfigurace domácího Kinect Mo-cap studia	34

A Příloha - DVD

DVD: Na přiloženém DVD se nachází zpracované nasnímané porovnávací a speciální sekvence ve formátu mp4, zdrojové kódy webu How To Mo-cap, vytvořené video tutoriály a aplikace Coin Catcher - Kinect v2.

B Příloha - Web

Web: <http://www.howtomocap.com/>

C Příloha - Youtube kanál

Youtube: <https://www.youtube.com/channel/UC0GgSyjP5ZVpYJCuwWqAPhQ>