



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

Diplomová práce

Bezdotykové ovládání interaktivních výukových aplikací s využitím technologie Leap Motion

Contactless control of interactive training applications using
Leap Motion technology

Vypracoval: Bc. Tomáš Svatek
Vedoucí práce: PaedDr. Petr Pexa Ph.D.

České Budějovice 2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš SVATEK**
Osobní číslo: **P13640**
Studijní program: **N7503 Učitelství pro základní školy**
Studijní obory: **Společný základ**
Učitelství fyziky pro 2. stupeň základních škol
Učitelství informatiky pro 2. stupeň základních škol
Název tématu: **Bezdotykové ovládání interaktivních výukových aplikací s využitím technologie Leap Motion**
Zadávací katedra: **Katedra informatiky**

Z á s a d y p r o v y p r a c o v á n í :

Cílem diplomové práce bude zpracovat možnosti využití nové technologie Leap Motion při bezdotykovém ovládání interaktivních výukových aplikací se zaměřením na výuky fyziky na základní škole. Diplomant prověří možnosti a kvalitu ovládání již hotových aplikací pro interaktivní tabule a otestuje dostupné aplikace vytvořené přímo pro technologii Leap Motion. Zjistí výhody a možné nevýhody této technologie porovnáním s klasickým dotykovým ovládáním interaktivních tabulí či tabletů a naprogramuje vlastní ukázkovou aplikaci, kterou otestuje ve vlastní pedagogické praxi. Výstupem práce bude mj. videozáznam ukázkové hodiny a vyhodnocení výzkumu, zaměřeného na možné využití Leap Motion ve výuce na základní škole a porovnání této technologie s technologiemi klasickými. Součástí práce bude i souhrn informací o této technologii (software, dostupné programy, podpora, finanční náročnost apod.).

Rozsah grafických prací: CD ROM

Rozsah pracovní zprávy: 60

Forma zpracování diplomové práce: tištěná

Seznam odborné literatury:

1. LEAP MOTION. Leap Motion : Leap Motion — Mac & PC Gesture Controller for Games, Design, & More [online]. 2013 [cit. 2013-11-07]. Dostupné z: <https://www.leapmotion.com/>
2. LEAP MOTION. Leap Motion: Leap Motion Developer [online]. 2013 [cit. 2013-11-07]. Dostupné z: <https://developer.leapmotion.com/>
3. UNITY TECHNOLOGIES. Unity 3D: Game engine, tools and multiplatform [online]. 2013 [cit. 2013-11-07]. Dostupné z: <http://unity3d.com/unity>
4. METODICKÝ PORTÁL RVP. /Metodický portál pro inspiraci a zkušenosti učitelů/ [online]. [cit. 2012-03-23]. Dostupné z: www.rvp.cz, <http://www.rvp.cz>
5. RAMBOUSEK, V. Funkce technických výukových prostředků ve vyučovacím procesu. In: Didaktická technologie. Praha : Karolinum, 1994, s. 42 - 90.
6. POKORNÝ, Martin./Digitální technologie ve výuce/. Vyd. 1. Kralice na Hané: Computer Media, 2009, 72 s. ISBN 978-80-7402-012-4.
7. Pecina, P. a kol. Metodika pro tvorbu a aplikaci didaktických prostředků propagujících vědu a techniku a profesní kariéru v rámci stávajících předmětů fyzika, chemie a technická výchova na základních školách. 1. vydání. Brno: PdFMU, 2009. 72 s. ISBN: 978-80-210-5088-4. [cit. 12. 6. 2011]. Dostupné z <http://www.didactex.cz/vystupy/obecnametodika.pdf>

Vedoucí diplomové práce:

PaedDr. Petr Pexa, Ph.D.

Katedra informatiky

Datum zadání diplomové práce:

11. listopadu 2013

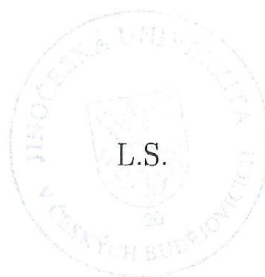
Termín odevzdání diplomové práce:

30. dubna 2015



Mgr. Michal Vančura, Ph.D.

děkan



doc. PaedDr. Jiří Vaníček, Ph.D.

vedoucí katedry

V Českých Budějovicích dne 11. listopadu 2013

Prohlášení

Prohlašuji, že svoji diplomovou 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 diplomové práce - a to v nezkrácené podobě - v úpravě vzniklé vypuštěním vyznačených částí archivovaných fakultou 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 dne

Podpis autora

Abstract

Práce pojednává v teoretické části o možnosti využití bezdotykové technologie Leap Motion ve výuce na základní škole pro ovládání interaktivních aplikací se zaměřením na výuku fyziky. Zjišťuje, jakým způsobem je možné ovládat již vytvořené aplikace, které nejsou programovány přímo pro bezdotykovou technologii, ale jsou určeny pro práci s interaktivní tabulí či klasickou prací na počítači, a jejich výhody a nevýhody. Zabývá se mimo jiné dostupností aplikací a nabízí souhrn informací o technologii Leap Motion a možnostech nahrazení interaktivních tabulí. Součástí práce je i popis technologie a možnosti vývoje vlastních aplikací. Cílem praktické části je pak naprogramovat vlastní výukovou aplikaci, která bude otestována v hodině a součástí bude i výzkum, který zjistí zájem učitelů o tuto technologii.

Klíčová slova:

Leap Motion, Unity 3D, bezdotykové ovládání, ovládání gesty, fyzika, interaktivní tabule, základní škola

Abstract EN

In its theoretical part, this thesis treats possibilities of using no-touch technology Leap Motion in elementary school lessons for operating interactive applications focused on physics education. The thesis investigates the ways in which it is possible to operate already published applications which are not programmed specifically for a no-touch technology and which are, however, intended for work with an interactive whiteboard or for a standard computer work. Their advantages and disadvantages are discussed as well. The thesis deals, among other things, with the availability of relevant applications and offers a summary of information about Leap Motion technology and of opportunities for replacing interactive whiteboards. The thesis also includes a description of the technology and of the potential for developing your own applications. The aim of the practical part is to create a new didactic application which will be tested in lessons. This part also includes a poll which will find out what is the interest in Leap Motion technology among teachers.

Keywords:

Leap Motion, Unity 3D, hands free controller, Gesture Controller, physics, interactive board, primary school

Poděkování

Rád bych tímto poděkoval svému vedoucímu práce panu PaedDr. Petru Pexovi, Ph.D. za rady a čas, který mi věnoval při řešení dané problematiky a jeho doporučení, jakož i za pevné nervy.

Dále také děkuji kamarádům, kteří svými nápady přispěli a v budoucnu ještě určitě přispějí k lepší funkčnosti naprogramované aplikace, jmenovitě pak Ing. Onřeji Nedvědovi, který mi pomohl s mými začátky s 3D grafikou.

Děkuji rovněž své rodině, jež se mnou měla trpělivost a ve studiu mne vždy podporovala.

Obsah

1	Úvod	10
1.1	Cíle práce	10
1.2	Metoda práce	12
2	Tehnologie Leap Motion	13
2.1	Historie	13
2.2	Technologie	14
2.2.1	Leap Motion SDK	15
2.2.1.1	Popis SDK	16
2.2.1.2	Leap Motion a matematika	18
2.2.2	Gesta	18
2.2.2.1	Krouživý pohyb	19
2.2.2.2	Zmáčknutí klávesy	20
2.2.2.3	Švihnutí	20
2.2.2.4	Ťuknutí na obrazovku	20
2.3	Budoucnost Leap Motion	21
3	Leap Motion jako nástroj ve výuce	24
3.1	Nebezpečí Leap Motion ve výuce	26
3.2	Dostupnost softwaru	26
3.2.1	Představení několika aplikací pro 2. stupeň ZŠ a SŠ	27
3.2.2	Aplikace použitelné v předškolním věku a na 1. stupni ZŠ	30
3.2.3	Leap Motion a vývojové prostředí Scratch	31
3.3	Možnost ovládání aplikací nejen pro interaktivní tabule	32
3.3.1	Touchless	32
3.3.2	Airinput Trial	34
3.3.3	Pointable	35
3.3.4	Mudra mouse	38
3.4	Shrnutí	39
3.5	Zájem učitelů o technologii Leap Motion	39
3.5.1	Vyhodnocení dotazníku	40
3.5.2	Shrnutí dotazníku	43

4	Praktická část - vlastní výuková aplikace	44
4.1	Téma aplikace	44
4.2	Popis aplikace a její cíle	44
4.3	Vývoj aplikace	45
4.3.1	Prostředí Unity 3D	46
4.3.1.1	Skriptování	49
4.3.1.2	Animace	49
5	Aplikace - Sluneční soustava	51
5.1	Interakce Leap Motion	51
5.2	Pohyb po obrazovce	51
5.3	Pohyb ve scéně	52
5.3.1	Interakce s uživatelským rozhraním a planetami	52
5.3.1.1	Kliknutí aneb zobraz panel pomocí Leap Motion	54
5.3.2	Pohyb po scéně	55
5.3.3	Ovládání gesty	56
5.4	Planety	57
5.4.1	Slunce	58
5.5	Scény	59
5.5.1	Intro	59
5.5.2	Hlavní scéna	61
5.5.3	Scéna porovnání	62
5.6	Kvíz	63
5.7	Zvukový doprovod	63
5.8	Export a distribuce aplikace	64
5.9	Alternativní výstup	65
5.10	Ověření aplikace	65
6	Závěr	66
7	Seznam použité literatury a zdrojů	67
8	Přílohy	74
8.1	Vybrané zdrojové kódy	75

8.1.1	Vykreslení kurzoru dle pozice ukazováčku	75
8.1.2	Pohyb ve scéně pomocí dlaně	79
8.1.3	Gesta	87
8.2	Dotazník	90
8.3	Vyhodnocení dotazníku	93
8.4	Příloha CD	98

1 Úvod

Moderní technologie jsou všude kolem nás a jejich vývoj jde stále dopředu. Není tedy divu, že i školy se snaží držet krok a nakupují moderní technologie ke zjednodušení a zábavnějšímu stylu výuky. Tablety a interaktivní tabule jsou dnes a denně ve škole častým slovem. Učitel se díky těmto technologiím snaží žáky motivovat a povzbudit ve výuce. Přinášená interaktivita je další faktor ve vedení hodiny oproti prostému sezení na židli. Tím, že žáci mění aktivity se pro ně výuka stává snesitelnější a zábavnější. Hodiny dostávají nový spád. Učitel může pomocí počítačové technologie simulovat, nakreslit, naměřit či jinak předvést věci, které by jinak s obyčejnou tabulí nemohl a největší výhodou je zapojení samotných žáků při řešení problémů pomocí moderní technologie.

Bohužel, nákup takovéto technologie není zrovna nejlevnější (řádově desetitisíce) a tudíž si ji nemůže dovolit jakákoliv škola. Právě cenová náročnost a zájem o nové technologie mě přiměl zaměřit se na to, zdali by se tato technologie nenechala nahradit levnější záležitostí, a to například Leap Motion kontrolerem, jehož cena se pohybuje kolem 2000 Kč, jež přináší větší interaktivitu a pohyb do výuky. Jedná se zařízení, které dokáže ovládat aplikace pomocí gest prováděných nad snímačem.

Jaká je ale efektivita ovládání aplikací, které nejsou napsány přímo pro tuto technologii? Bude děti bavit ovládat počítač pohybem rukou ve vzduchu? Budou se děti opravdu učit z aplikace, nebo zabere více času anučit se ovládat aplikaci? Měli by učitelé vůbec zájem o takovou technologii? Na tyto a další otázky se pokusím odpovědět právě v této diplomové práci.

1.1 Cíle práce

Největší překážkou zavádění nových technologií do škol jsou finance. Školy se proto snaží hledat alternativy k dnes nabízeným produktům a leckdy staví cenu před kvalitu učebních pomůcek. Leap Motion je relativně nová technologie, která se stále vyvíjí a která mě zaujala nejen cenou tzv. kontroleru, ale i možnostmi které toto malé zařízení nabízí, stejně jako počtem dostupných aplikací v Leap App store¹

¹<https://apps.leapmotion.com/>

Cíli této práce tedy je:

- posouzení možnosti využití technologie Leap Motion na školách jako možnou náhradu za interaktivní tabule
- vyzkoumání možností ovládání standardních i interaktivních aplikací, které nejsou psány přímo pro tuto technologii
- popis možností a technologie Leap Motion
- vyhledání a popsání informací o dostupnosti softwaru pro Leap Motion
- naprogramování vlastní výukové aplikace s manuálem
- natočení použití výukové aplikace v hodině fyziky
- zjištění názorů učitelů na použití technologie Leap Motion ve výuce

Vzhledem k zaměření práce do výuky fyziky, bude i naprogramovaná ukázková aplikace do tohoto předmětu. Vybral jsem si téma "Sluneční soustava". Podobná aplikace na Leap App storu již existuje, ale bohužel pouze na operační systém MacOS. Aplikace poskytne žákovi možnost "proletět" se po naší sluneční soustavě. Od Slunce po Pluto a kousek dál. Tato aplikace nabídne popis jednotlivých planet, jejich simulaci rotace nejen kolem Slunce, ale i rotace kolem jejich osy a měsíců okolo planet. Planety bude možno zvětšit a urychlit čas, takže se žák může podívat, jaký je pohyb planet v čase. Samozřejmě bude možnost pozorování pohybu planet a zkoumání například vrženého stínu měsíce na zemi. Aplikace bude obsahovat jednoduchý kvíz, jehož odpovědi budou žáci možni najít v informacích o planetách.

Celá aplikace bude programována v prostředí Unity 3D. Jedná se o herní engine. Výsledná aplikace bude dostupná online pro desktop, s ovládáním jak Leap Motion tak i pomocí myši, a pro web pouze s ovládáním myši.

1.2 Metoda práce

Práce byla rozdělena na několik etap. V první se jedná sepsání informací o technologii Leap Motion a to nejen historie, ale i dostupnost softwaru a jejího budoucího vývoje.

Druhá etapa se bude týkat pročítání tutoriálů, online zdrojů a dokumentace, aby bylo možné naprogramovat výukovou aplikaci v prostředí Unity 3D s bezdotykovým ovládáním Leap Motion. S tím bude souviset vytvoření myšlenkové mapy pro rozvrhnutí aplikace a popisu jejích funkcí. Ta se v průběhu vývoje aplikace měnila s přibývajícím problémy a požadavky.

Bude následovat sepsání teoretické části a navázání na praktickou část, která bude z teoretické vycházet. Aplikace bude postupně upravována do finální podoby.

Po zpracování praktické části je tuto nutné otestovat ve výuce na základní škole. Zde se natočí video z výuky, které bude součástí elektronické přílohy na DVD a bude proveden průzkum, zdali by učitelé měli o tuto technologii zájem.

Počítá se s nabídnutím hotové aplikace na oficiální obchod Leap Motion Apps Store ke stažení zdarma jako výukový materiál.

2 Technologie Leap Motion

Leap Motion je technologie, která nám umožňuje ovládat aplikace pomocí pohybů celé ruky, dlaně, čí prstů. To vše dokáže zařídit malé zařízení, které je o něco větší než flash disk.



Obrázek 1: Leap Motion [3]

Sama firma má vlastní internetový obchod s aplikacemi. Jedná se o Leap App Store neboli Airspace. Ke dni 28. 03. 2015 obsahuje kolem 220 aplikací. Zároveň klesla i cena zařízení. Dnes se pohybuje kolem \$90. Mohlo by se zdát, že konkurencí by mohl být Kinect. Nicméně díky velikosti, je Leap Motion lépe přenosný a má citlivější rozlišení.

2.1 Historie

Společnost Leap Motion byla založena v roce 2010 a to Michaelem Buckwaldem a Davidem Holzem, nicméně samotná technologie spatřila světlo světa již v roce 2008. Její vynálezce byl právě David Holz, který tehdy studoval v doktorandském studiu na Univerzitě v Severní Carolíně v Chapel Hill. Při práci s 3D grafikou ho rozčilovala doba potřebná k vymodelování objektu, stejně jako omezení práce s myší a klávesnicí.

Strávil téměř pět let vývojem ovladače, který je schopen určit pohyby rukou, jejich směr, rychlost, rotaci i jednotlivé prsty. V roce 2012, 21. května, firma oficiálně oznámila vydání zařízení pod jménem *The Leap*.

V říjnu téhož roku byl uvolněn i software umožňující vývoj aplikací, a zároveň s tím bylo dáno do distribuce 12 000 kusů kontroleru pro vývojáře, kteří projevíli zájem o vývoj aplikací právě pro zařízení *The Leap*. Kvůli kritice se však zpozdilo oficiální prodej a místo března roku 2013, se začal plně prodávat veřejnosti až od června téhož roku. Na internetovém portálu TechCrunch², zabývajícím se technologickými novinkami, bylo oznámeno, že se prodalo kolem 500 000 kusů zařízení, což bylo méně, než firma očekávala. V květnu 2014 vydala firma software pro zařízení ve verzi 2. K dnešnímu dni je verze softwaru 2.2.4.[1]



Obrázek 2: Leap Motion - historický vývoj [7]

2.2 Technologie

Jedná se o malé vstupní USB zařízení, které bylo původně navrženo k umístění na stůl. Jako snímače používá dvě monochromatické infračervené kamery a tři infračervené LED diody. Zařízení je schopné snímat pohyb rukou a to v prostoru o tvaru polokoule do vzdálenosti cca 70 cm. V této ploše diody generují 3D vzorek z teček z infračerveného

²<http://techcrunch.com/>

světla³. Kamera poté snímá až 300 snímků za sekundu a vyhodnocuje data na základě přerušení a odrazu. Ta jsou přenášena pomocí USB kabelu do počítače, kde je software analyzuje. To se děje za pomoci komplexních matematických operací. Ty si společnost dobře chrání a vývojář má, díky SDK⁴, přístup k metodám, které vrací vyhodnocené výsledky. Navíc se jedná o syntetizaci 3D obrazu z 2D generovaných snímků.



Obrázek 3: Leap Motion - rozebraný [11]

2.2.1 Leap Motion SDK

Firma Leap Motion vydala již několik verzí softwaru ke kontrolieru. S tím se pojí i software pro vývoj aplikací tzv. SDK. Toto SDK obsahuje potřebné knihovny, v nichž jsou zaneseny metody pro práci s přichozími daty. Firma se snaží, aby zařízení bylo možno používat v co největší škále možností a proto jsou vydávány SDK pro řadu programovacích jazyků a vývojových prostředí.



Obrázek 4: Leap Motion - podporované jazyky a vývojová prostředí [4]

Stačí knihovny pro konkrétní jazyk nainportovat do projektu. Poté se zpřístupní všechny důležité metody. SDK je velice dobře zdokumentováno [3]. Já osobně jsem používal SDK pro Unity 3D[4] a C#. Ke každé verzi je vydán i seznam změn.

³https://www.youtube.com/watch?v=UI5EBzU_QqM

⁴Software Development Kit

2.2.1.1 Popis SDK

SDK nám umožňuje pomocí API⁵ přistupovat právě k oněm metodám a informacím. Já zde ukáži jen krátkou ukázkou. Více se tomuto tématu budu věnovat při popisu vytvořené aplikace. Navíc jsou dostupné metody podrobně popsány v dokumentaci. Jak jsem zmiňoval výše, je nutné nejprve knihovny přepokopírovat k projektu, poté je nainportovat do jednotlivých tříd. První se musí vytvořit objekt kontroleru.

```
using System;
using Leap;      //import knihovny LEAP
using UnityEngine;
using UnityEngine.UI;

public class MovingCameraByLeap: MonoBehaviour
{

    Controller controler;

    void Start(){
        controler = new Controller (); //vytvoření nového kontroleru
    }
}
```

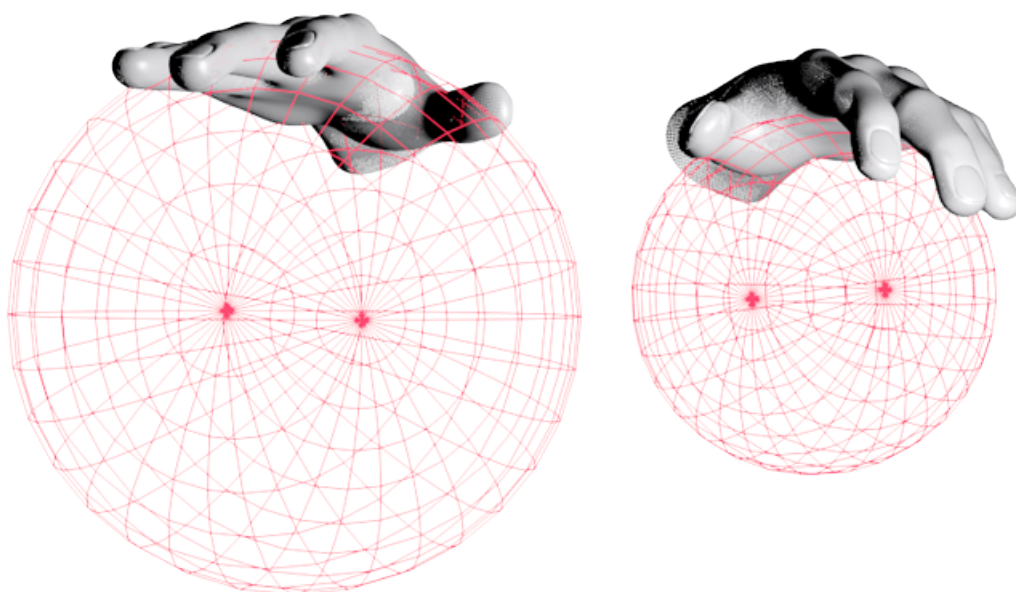
Z objektu kontroleru pak můžeme přistupovat k dalším metodám. Následující řádky nám například umožní zjistit, kolik rukou je v daném snímku, jak vytvořit objekt ruky a zjistit, zdali je levá nebo pravá.

```
int handCount = controler.Frame().Hands.Count;
Hand hand = controler.Frame().Hands[1];
if (hand.IsLeft){
    print("Ruka je levá");
}
```

⁵Application interface

Pokud máme objekt ruky, můžeme zjistit, jakým směrem se pohybuje za pomoci procházení historií snímků. Můžeme zjistit, jestli, kolik a jaké prsty jsou zavřené v dlani nebo ne. Samozřejmostí je, že veškeré části ruky se dají definovat jako objekt. Objekt ruky poté obsahuje další metody například:

- pro práci s transformační maticí
- se zjištěním směru pohybu dlaně
- sílu stisku
- jestli je dlaní dolů nebo ne
- velikost dlaně
- poloměr koule kterou ruka opisuje
- ...

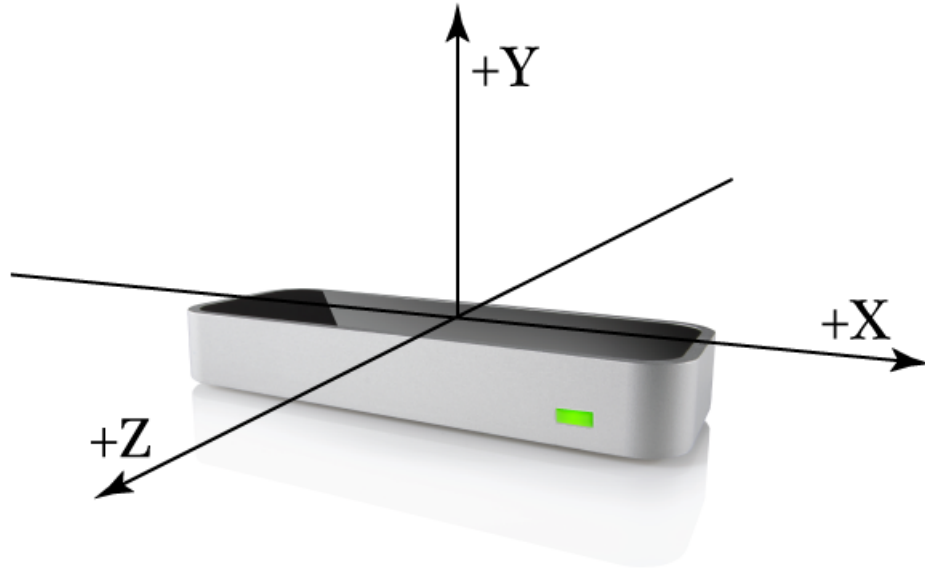


Obrázek 5: Leap Motion - poloměr koule kterou ruka opisuje [4]

Kromě ruky umožňuje Leap Motion snímání a ukazování pomocí objektů, například propisky, pro lepší a přesnější ukazování. Díky dostupným informacím si programátor může omezit funkce programu podle počtu vztyčených prstů, počtem rukou ve snímku, vzdáleností od kontroleru nebo velikostí koule kterou opisuje ruka, rychlostí jejího pohybu, jestli je zavřená nebo otevřá dlaň, atp.

2.2.1.2 Leap Motion a matematika

Při vývoji aplikace je nutné si uvědomit, že leap motion využívá komplexní matematické operace. Pohybujeme se ve 3D prostoru a proto je nejjednodušší používat vektory a transformační matice k popisu pohybu ruky nebo předmětu.

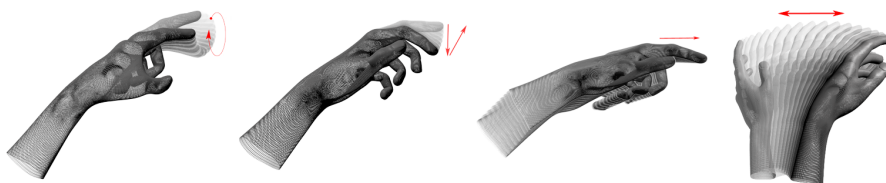


Obrázek 6: Leap Motion - směr os [2]

2.2.2 Gesta

Jednou vestavěnou funkcí je rozpoznávání základních gest a to:

- krouživý pohyb prstem (Circle gesture)
- zmáčknutí klávesy - cvrknutí bez použití palce (Key tap gesture)
- ťuknutí na obrazovku (Screen tap gesture)
- švihnutí ruky (Swipe gesture)



Obrázek 7: Leap Motion - gesta: kroužení, stiknutí klávesy, ťuknutí, švih [4]

Každé gesto má možnost nastavení základních parametrů. Pro použití v aplikaci se musí gesta povolit. K tomu slouží následující řádky. Každé gesto se musí povolit zvlášť. Ta jsou poté rozpoznána, jsou-li provedena kdekoli ve snímacím prostoru kontroleru.

```
//povolení gest
controller.EnableGesture(Gesture.GestureType.TYPE_CIRCLE);
controller.EnableGesture(Gesture.GestureType.TYPE_KEY_TAP);
controller.EnableGesture(Gesture.GestureType.TYPE_SCREEN_TAP);
controller.EnableGesture(Gesture.GestureType.TYPE_SWIPE);
```

2.2.2.1 Krouživý pohyb

Tato třída reprezentuje krouživý pohyb prstem. Ten může nabývat tří stavů:

- STATE_START – krouživý pohyb začal a pokračoval dostatečně dlouho, aby byl vyhodnocen jako kruh
- STATE_UPDATE – pohyb pokračuje
- STATE_STOP – gesto bylo dokončeno

U toho gesta můžeme nastavit následující parametry, které se musejí uložit:

- minimální rádius terý musý být proveden
- minimální obsaný úhel

```
ontroller.Config.SetFloat("Gesture.Circle.MinRadius", 10.0f);
controller.Config.SetFloat("Gesture.Circle.MinArc", .5f);
controller.Config.Save();
```

2.2.2.2 Zmáčknutí klávesy

Tato třída reprezentuje simulaci zmáčknutí klávesy. Dříve než je toto gesto rozpoznáno, musí se prst an chvíli zastavit a poté začít provádět gesto. U toho gesta můžeme nastavit následující hlavní parametry:

- minimální rychlost směrem dolů (k dlani)
- minimální uraženou vzdálenost

```
controller.Config.SetFloat("Gesture.KeyTap.MinDownVelocity", 40.0f);  
controller.Config.SetFloat("Gesture.KeyTap.HistorySeconds", .2f);  
controller.Config.SetFloat("Gesture.KeyTap.MinDistance", 1.0f);  
controller.Config.Save();
```

2.2.2.3 Švihnutí

Tato třída reprezentuje simulaci švihnutí prstem nebo nástrojem. U toho gesta můžeme nastavit následující hlavní parametry:

- minimální uražená vzdálenost
- minimální rychlost

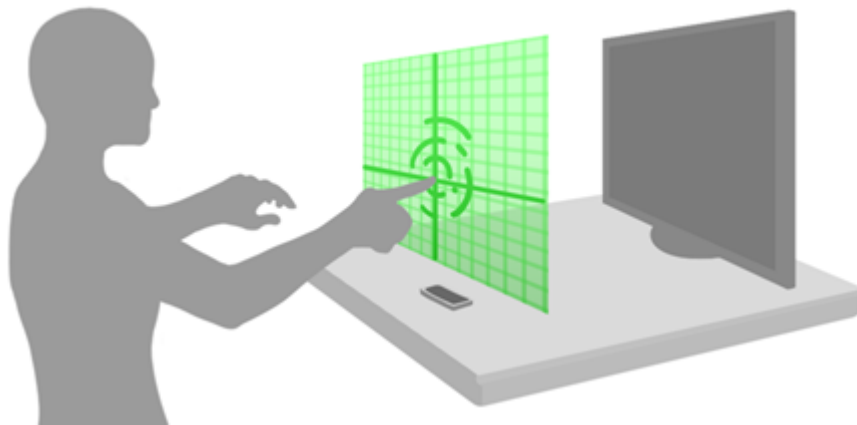
```
controller.Config.SetFloat("Gesture.Swipe.MinLength", 200.0f);  
controller.Config.SetFloat("Gesture.Swipe.MinVelocity", 750f);  
controller.Config.Save();
```

2.2.2.4 Ťuknutí na obrazovku

Tato třída reprezentuje simulaci ťuknutí do obrazovky. Základem je, že se prst musí pohnout dopředu a poté se vrátit přibližně do výchozí polohy. Opět musí nastat krátká pauza před provedením gesta. U toho gesta můžeme nastavit následující hlavní parametry:

- minimální rychlost směrem dopředu (k obrazovce)
- minimální uraženou vzdálenost

```
controller.Config.SetFloat("Gesture.ScreenTap.MinForwardVelocity",
    30.0f);
controller.Config.SetFloat("Gesture.ScreenTap.HistorySeconds", .5f);
controller.Config.SetFloat("Gesture.ScreenTap.MinDistance", 1.0f);
controller.Config.Save();
```



Obrázek 8: Leap Motion - gesta: ťuknutí [9]

Každé gesto samozřejmě obsahuje další metody pro zjištění dodatečných informací, jako je doba trvání provádění gesta, jestli je gesto validní, ruku, která gesto provádí, výchozí pozici odkud je gesto prováděno, atp. Vše je opět popsáno v dokumentaci k SDK.

2.3 Budoucnost Leap Motion

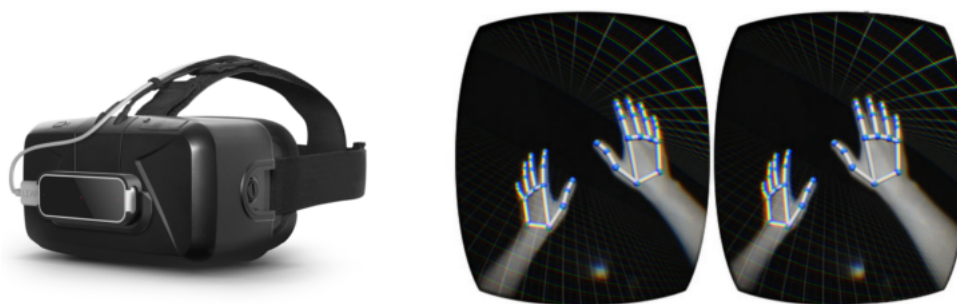


Obrázek 9: Leap Motion - budoucnost? [9]

Chvilí po vydání technologie Leap Motion a chvilkovém testování někteří recenzenti tuto technologii odsoudili k záhubě. Je pravda, že první verze softwaru byla tak trochu „nehorabaná“. Každý kdo to zkusil musel uznat, že ovládání občas pokulhávalo. Jednalo

se o horší detekci částí ruky. Občas mizely prsty a to způsobovalo problém v ovládání aplikace. Ale i tak se s kontrolerem nechaly dělat divy. Například můžete ovládat známý prezentační software Prezi pomocí Leap Motion skoro jako Tom Cruise ovládal počítačové rozhraní ve filmu *Minority Report*[10]. Nebo proč neovládat dálkově roboty⁶? Bylo by možné ovládat pohybem ruky robota pro zneškodnění bomb daleko jemněji?

Nicméně firma se nevzdala a software zlepšovala. Dnes, jak již bylo zmíněno, máme verzi softwaru 2.2.4. Ta přinesla několik zlepšení včetně lepší detekce prstů a doplňujících metod pro práci s objekty. Díky tomu se tato technologie spojila s technologií Oculus Rift⁷[12]. Byl dokonce uvolněn i kód pro *Planetarium*[14].



Obrázek 10: Leap Motion - Leap Motion a Oculus Rift [9]

Dalším projektem s Oculus Rift je tzv. OSVR⁸. Otevřená platforma pro stavbu vaší vlastní virtuální reality. To podporují známe firmy jako:

- Sensics – výrobce brýlí pro virtuální realitu - <http://sensics.com/>
- Razer – výrobce herního hardwaru a softwaru - <http://razerzone.com/>
- Gearbox software – softwarová vývojová firma - <http://gearboxsoftware.com>
- Sixense – výrobce hardwaru k virtuálním brýlům (ovladač STEM)
- <http://sixense.com>
- Leap Motion – Výrobce bezdotykového kontroleru - <http://leapmotion.com>

⁶<https://www.youtube.com/watch?v=sBzXRc9v7PU>

⁷<https://www.oculus.com/>

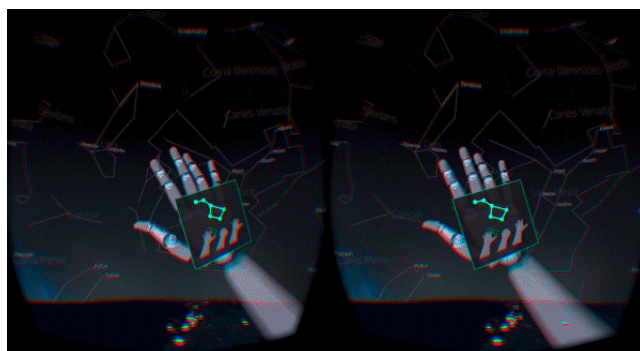
⁸Open Source Virtual Reality - <http://www.osvr.com/index.html>



Obrázek 11: Leap Motion - OSVR [13]

Momentální nevýhodou Leap Motionu je kabel, který potřebujete. Lidé z komunity jsou však vynalézaví a snaží se propojit konroler s technologií bluetooth⁹. To by v budoucnosti mohlo přinést větší volnost pohybu a to by mělo obrovskou výhodou nejen ve školství ale i herním průmyslu.

Virtuální a rozšířená realita je dnes na vzestupu, což mohou dokládat i konkurenční brýle od Microsoftu Hololens¹⁰, které nejsou přímo virtuální realitou, ale onou rozšířenou realitou.



Obrázek 12: Leap Motion - Oculus Rift a planetárium[14]

⁹<http://cs.wikipedia.org/wiki/Bluetooth>

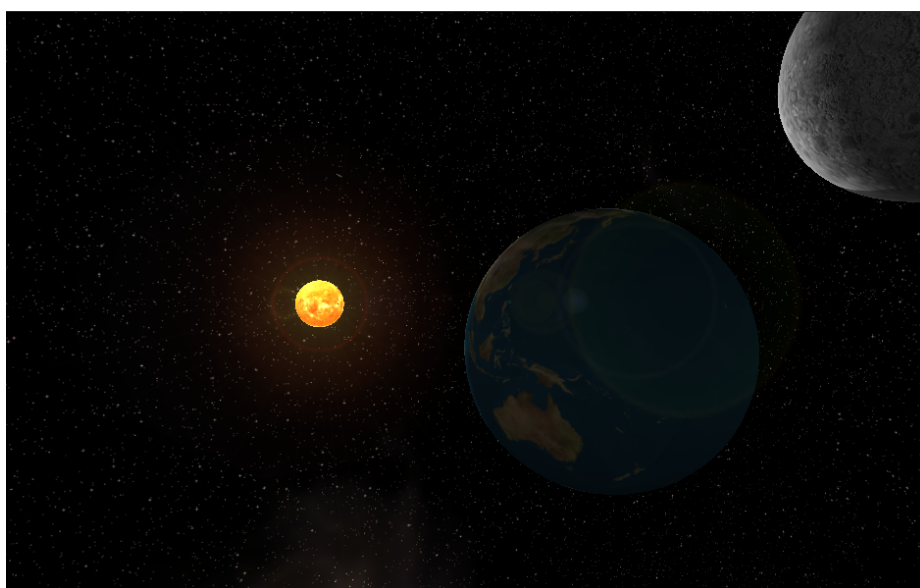
¹⁰<http://www.microsoft.com/microsoft-hololens/en-us>

3 Leap Motion jako nástroj ve výuce

V úvodu bylo zmíněno, že se v dnešní době nasazuje moderní technologie do školství, abychom zabavili děti, zkusili je motivovat a zároveň pobavit tak aby se i přes zábavu naučili to co měli. Emoce jsou nejsilnější vyvolávač paměti. Žák si spíše vzpomene na to, jak se bavil při pokusu něco ovládat či virtuálně rozbít, než že seděl celý den na židli a vůbec jeho jediný pohyb byl o přestávce na záchod.

S tím souvisí i doporučená změna aktivit ve výuce. Interaktivní tabule tehdy přišly jak na zavolanou. Bohužel, spousta učitelů nedokáže využívat software dodávaný s tabulí dot efektivně na to, aby dokázali vytvářet zábavné aplikace. Na druhou stranu zahrát si pexeso, poskládat správné odpovědi, nebo si zahrát riskuj je dobré rozptýlení a plní účel. Navíc s dodávaným softwarem není, pokud učitel pronikne do tajů vývoje aplikací, veliký problém si takové jednoduché hry vytvořit. Profesionálně připravených hodin je stále málo. Navíc je rozdíl mezi ActiveBoardem a SmartBoardem. Ale je interaktivní tabule dost zábavná? Samozřejmě existují animace, ale spousta z nich reaguje na stisk myši a táhnutí nic co by se nedalo udělat přes počítač tak v čem je výhoda?

Děti se dostanou z lavic, tabule je schopna odchyťávat několik vstupů najednou, takže v podstatě nahrazuje multiplayer. Navíc je interaktivní tabule vhodnější do určitých předmětů. Třeba čeština a pravopis, zeměpis, prvouka, fyzika, matematika Vždycky nacházíme nějaké ale. V tomto případě se naskytuje otázka: *Nešlo by to ještě zábavněji a s větší interaktivitou?*



Obrázek 13: Aplikace Sluneční soustava pro Leap Motion

Je pěkné, že si můžete popohnat animaci například štěpení atomu, ale co kdyby žák byl ten, kdo celou reakci inicializuje? Co kdyby bylo možné simulovat fyzikální jevy, jako například působení síly tak, jak to opravdu odpovídá skutečnosti? Co kdyby bylo možné se projít po historickém bojišti a přehrát nasimulované historické události, či se prolétnou skrz naší Sluneční soustavou?

To je to v čem vidím silnou stránku Leap Motion kontroleru. Možnost pohybovat se ve 3D světě pomocí zapojení dalších smyslů, než jen táhnutím elektornického pera. Navíc díky gestům by se nechali ovládat i roboti, takže by bylo možné např. propojení Lega®Mindstorm¹¹ s kontrolerem.

Dále jsem zmiňoval výhodu, kterou by přinesla technologie bluetooth v zařízení Leap Motion. Psané aplikace by mohly být pro více hráčů (uživatelů) a s napsaným softwerem pro omezení přístupu pro konkrétná zařízení by bylo možno, aby učitel předával ovládání jednotlivým zařízením podle jeho potřeby.



Obrázek 14: Leap Motion - Ukázka ovládání

Navíc díky jemným pohybům by zařízení mohlo učit jemné motorice. To by se kromě školství nechalo využít i u rehabilitačních programů.

¹¹<http://www.lego.com/en-us/mindstorms/>

3.1 Nebezpečí Leap Motion ve výuce

Všechno to zní moc hezky, ale ne vždy všechno jde tak jak si přejeme a i tato technologie není výjimkou. V čem by tedy mohlo tkvít nebezpečí? Bohužel ke komplexnosti 3D světa je náročné navrhnout ovládání aplikace. To by mělo být co nejintuitivnější a nejjednodušší, alespoň co by se nasazení ve výuce týkalo. A zde vidím velký problém.

Je pravděpodobné, že se děti zaměří spíše na to, jak ovládat aplikaci, než aby aplikaci sledovaly. V tom případě by měl čas strávený k naučení ovládání aplikace být vykompenzován množstvím informací, které si student odnese poté, co se naučí aplikaci ovládat. Výhodou je, že dnešní generace si rychle zvyká na nové technologie a dokáží se rychleji naučit zacházet s novými věcmi.

Většina aplikací navíc disponuje počátečním tutoriálem, který uživatele naučí vše co potřebuje od polohy ruky až po rychlosti, gesta, stisknutí tlačítka či uchopení věci.

3.2 Dostupnost softwaru

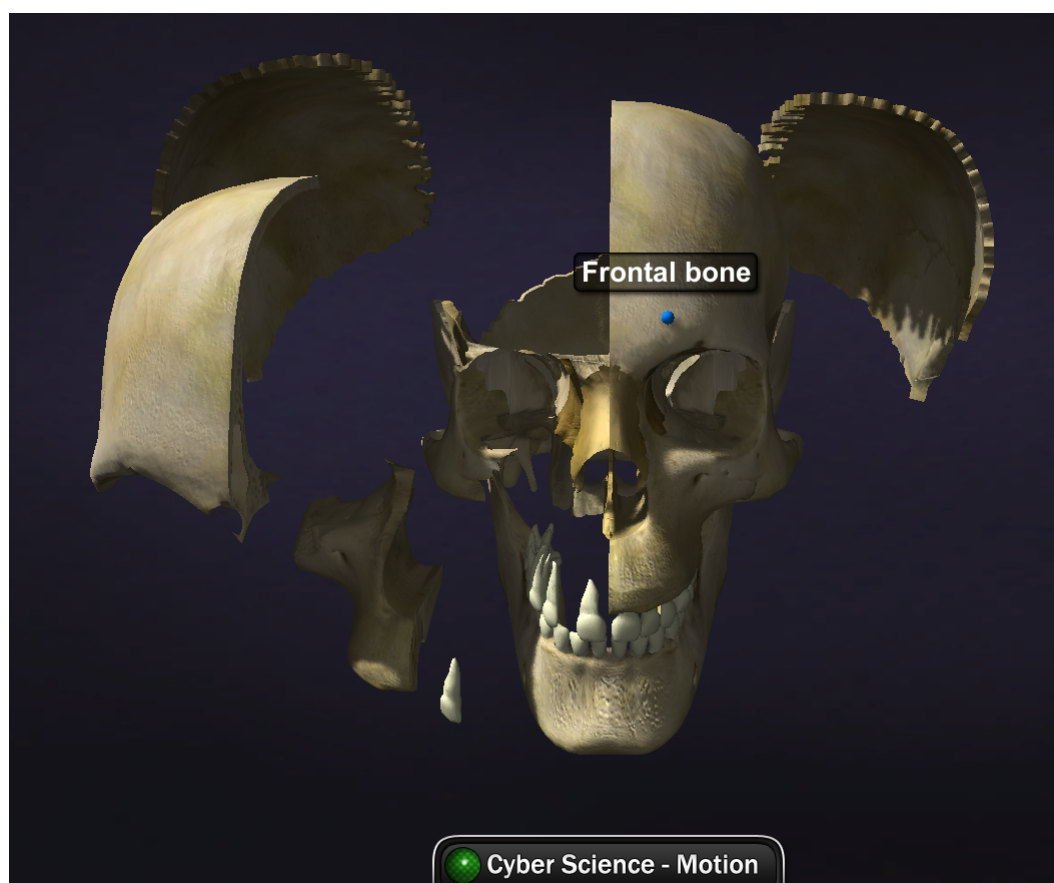
V úvodu bylo zmíněno, že Leap Motion disponuje vlastním obchodem pro šíření aplikací ať už zdarma, nebo za poplatek. Některé jsou určeny jen pro daný operační systém jiné jsou přístupné díky možnosti psát program v Unity, JavaScriptu nebo Unreal Engine. Samotný obchod obsahuje momentálně zhruba 220 aplikací ze všech možných kategorií:

- Ovládání počítače
- Kreativní nástroje
- Výuka
- Experimentální
- Hry
- Hudba a zábava
- Produktivita a utility
- Věda
- Virtuální realita

Já se v tuto chvíli zaměřím na výběr několika aplikací, které stojí za zmínku. A to převážně z oblasti výuky. Nevýhoda většiny aplikací je jazyková mutace. Nenarazil jsem na aplikaci, která by byla v jiném jazyce, než v angličtině.

3.2.1 Představení několika aplikací pro 2. stupeň ZŠ a SŠ

Rád bych vyzdvihl několik aplikací o kterých si myslím, že mají naději stát se výukovými. První se nazývá *Cyber Science - Motion*. Zabývá stavbou kostry, konkrétně lebky. Pomocí Leap Motion ovladače můžete lebku otáčet v prostoru, přibližovat, oddalovat a rozebírat na jednotlivé kosti, či si otestovat, jak rychle dokážete naopak kosti složit dohromady.



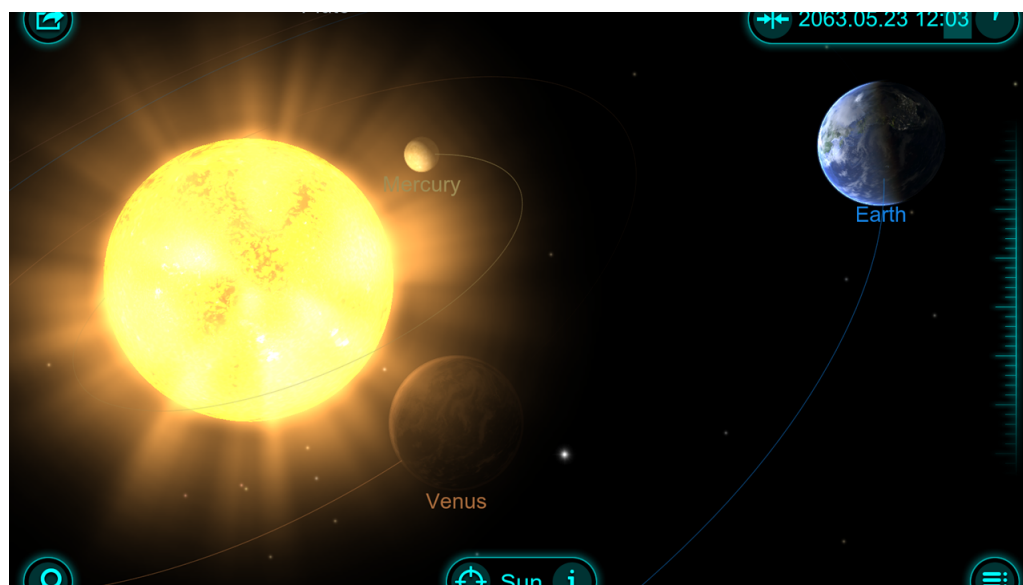
Obrázek 15: Cyber Science - Motion

Další aplikace *Form & Function 3D* se zabývá oběhovým systémem konkrétně zvířecích srdcí. To můžete zprůhlednit, natáčet a sledovat cirkulaci krve. Vše má samozřejmě popisky. Třetí aplikace je od stejného vývojáře jako první a to *Cyber Science - Motion Zoology*, ale tentokrát se jedná o rozebírání hmyzu. Ve verzi zdarma se jedná o terantuli.



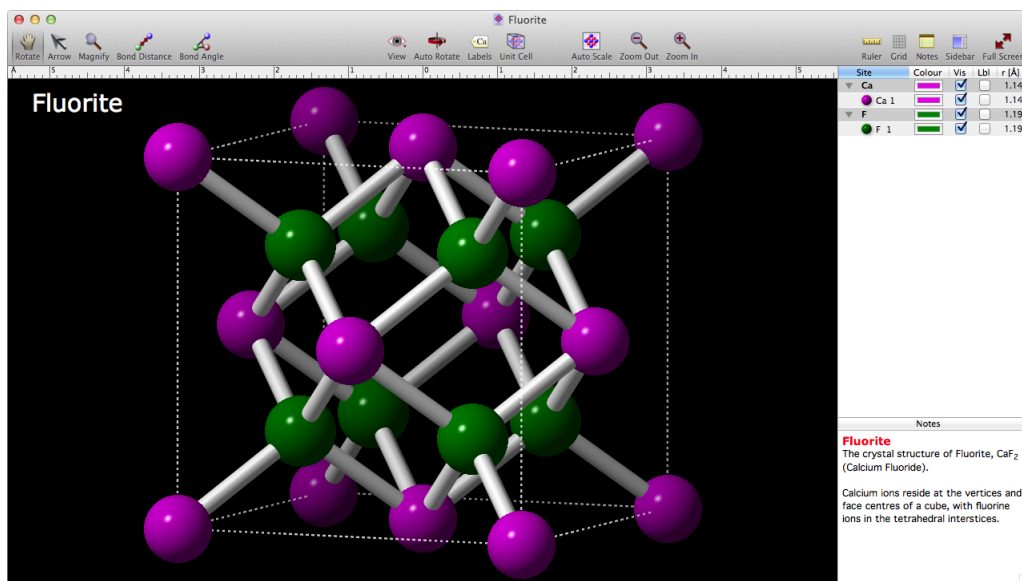
Obrázek 16: Form & Function 3D a Cyber Science - Motion: Zoology

Následující aplikace se zabývá Sluneční soustavou. Bohužel je dostupná pouze pro operační systém MacOS. To také vedlo ke zvolení tématu mé praktické části. Aplikace nese název *Solar Walk - 3D Solar System model*. Vizuálně je krásně zpracována. Doplněna o informace o jednotlivých planetách a krásnými efekty Slunce. Cena je €4,59.



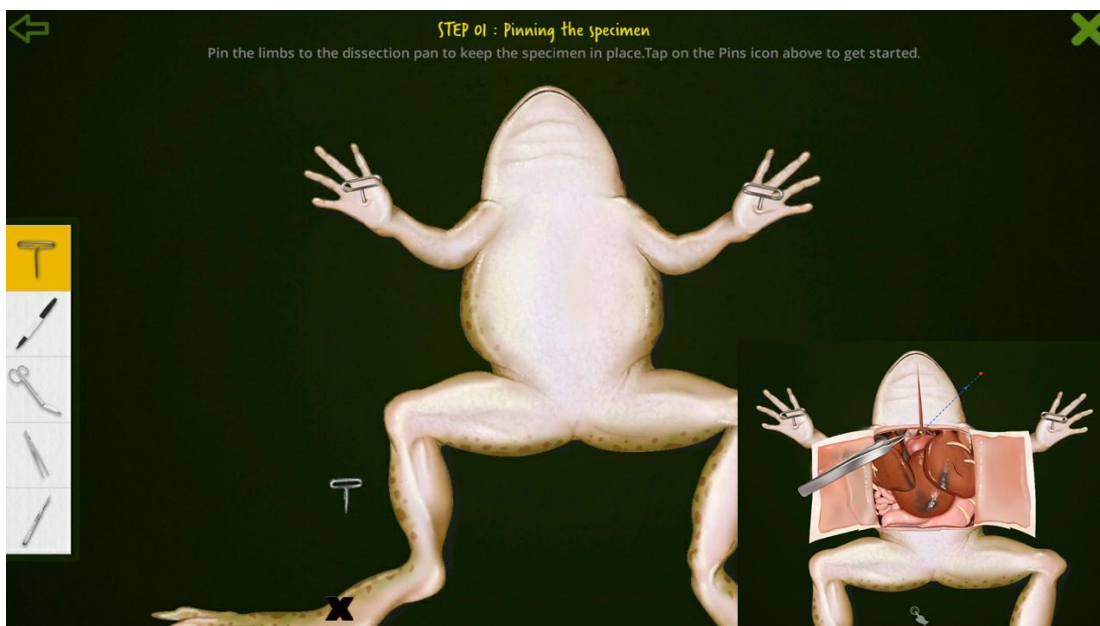
Obrázek 17: Aplikace Solar Walk[15]

V pořadí pátá aplikace se jmenuje *CrystlViewer*. Je zdarma a má za účel znázornit krystalické mřížky různých látek a chemických sloučenin. Bohužel je tato aplikace dostupná pouze pro MacOS. Jednotlivé modely mřížek můžeme otáčet. Také umožňuje měřit vzdálenosti jednotlivých atomů od sebe.



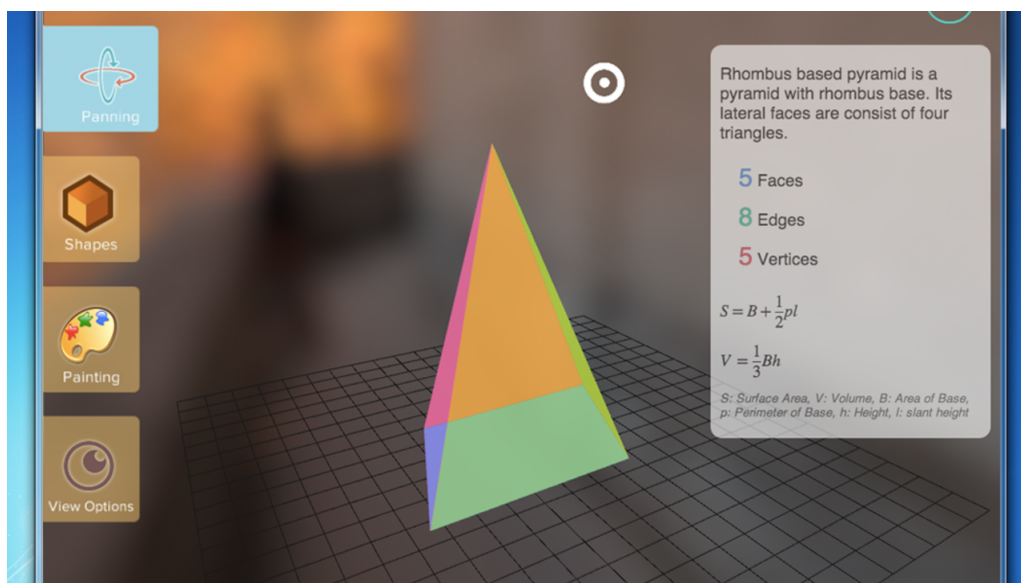
Obrázek 18: Aplikace CrystalViewer[16]

Předposlední aplikací pro vyšší ročníky je pitva žáby. Aplikace nese jméno *Frog Dissection* a umožňuje připíchnout žábu k pracovnímu stolu a provést pitvu bez trápení na reálném vzorku. Aplikace je dostupná pro MS Windows i MacOS a stojí €3,59.



Obrázek 19: Frog Dissection[17]

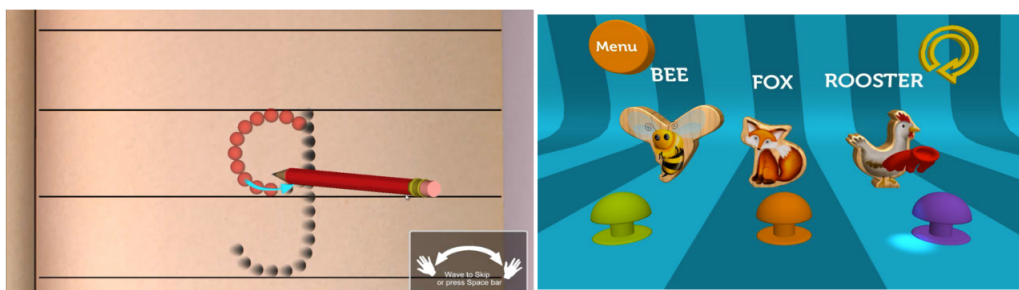
Poslední aplikací kterou bych rád zmínil je *3D Geometry*, která stojí €2,69 a je dostupná pro MS Windows i MacOS. Zde žáci budou moci manipulovat s 3D geoterickými objekty. Rozkládat je a pozorovat síť objektu. Jednotlivé strany si mohou obarvit různou barvou. Dále jsou k dispozici vzorečky pro výpočet obsahů a objemů jednotlivých těles.



Obrázek 20: 3D Geometry[18]

3.2.2 Aplikace použitelné v předškolním věku a na 1. stupni ZŠ

Kromě výše zmíněných aplikací, bych rád poukázal i na další, které jsou vhodné pro první stupeň základní školy. První je *Skywriting Alphabets*. Jedná se o psaní a učení písmen abecedy buď pomocí prstu nebo nástroje. Cena je €1,79. Druhá aplikace nese název *Curious Kids* a má za úkol naučit děti zvukům a počtům. Bohužel pro nás jsou tyto aplikace poze v anglické lokalizaci.

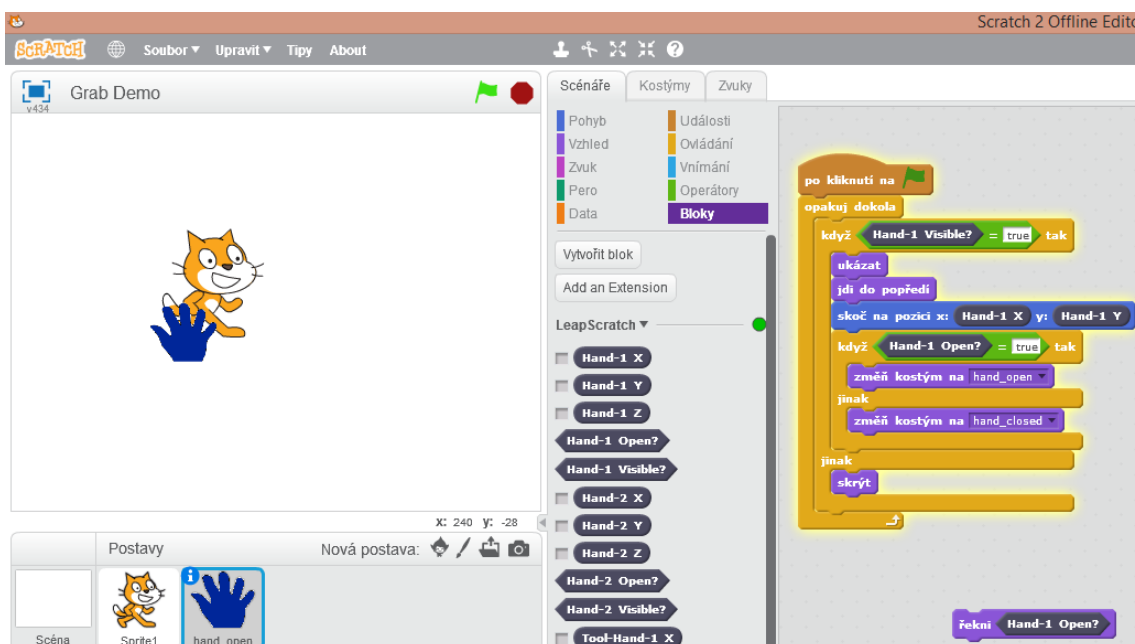


Obrázek 21: Skywriting Alphabets a Curious Kids[19][20]

3.2.3 Leap Motion a vývojové prostředí Scratch

Scratch¹² je oblíbené, moderní a jednoduché vývojové prostředí, jež umožňuje vytvářet i složité aplikace. Díky tomu, že se žáci nemusejí zabývat syntaxí, jako je tomu u jiných programovacích jazyků, a díky své jednoduchosti skládání příkazů, dostupnosti jednotlivých ovládacích prvků jako jsou cykly a podmínky, přehlednosti a barevnosti je toto prostředí hojně využíváno právě ve výuce informatiky pro rozšiřování schopností žáků řešit problémy okolo nich. To, a nejspíše možnost nabídnout i žákům základních škol možnost vytvořit si vlastní aplikaci pro bezdotykové ovládací zařízení Leap Motion, vedlo Krega Hanninga¹³ k vytvoření doplňku pro offline Scratch editor.

První co se musí udělat je stáhnout knihovnu napsanou v JSONu¹⁴, která se, po podržení klávesy SHIFT a kliknutí na SOUBOR a zvolení nabídky *Import experimental HTTP extension*, musí nainportovat. Poté se v nabídce bloky zobrazí příkazy právě pro práci s Leap Motion. Na stránce uvedené v poznámce pod čarou se nacházejí i ukázkové aplikace.



Obrázek 22: Leap Motion - rozšíření pro Scratch editor

Existuje spousta dalších aplikací. Výše zmíněné jsou pouze vybrané, které mě zaujaly a které jsem vyzkoušel

¹²<https://scratch.mit.edu/>

¹³<http://khanning.com>

¹⁴JavaScript Object Notation

3.3 Možnost ovládání aplikací nejen pro interaktivní tabule

V podkapitole výše jsme si ukázali nějaké aplikace, které jsou programovány přímo pro technologii Leap Motion. Ale jak je to s již existujícími aplikacemi? Naštěstí se předpokládalo, že by Leap mohl kompletně ovládat počítač. To vedlo k napsání hned několika softwarů. Ty jsou dostupné v Leap App storu. Některé jsou placené, jiné zdarma. Jak už to tak bývá, ne vždycky aplikace zdarma je nejhorší a ne vždy je placená aplikace nejlepší. Bylo otestováno několik základních aplikací, a to na již hotových programech, jakož to na celém systému. Co se týče aplikací pro interaktivní tabule, je nutné mít software k nim dodávaný a to Promethean ActiveInspire pro ActiveBoard a SMART Classroom pro SmartBoard.

- Touchless - zdarma
- Airinput trial - zdarma (*premium* €4,59)
- Pointable - €4,59
- Mudra Mouse - €4,59

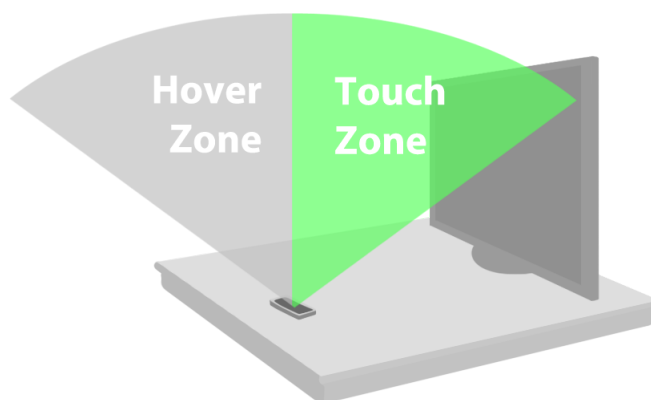
3.3.1 Touchless



Obrázek 23: Touchless for Windows[22]

Tato aplikace je zdarma a je z dílny Leap Motion. Princip fungování je v rozdělení snímané plochy nad kontrolerem na dvě části a to tzv. *Hover area* a *Touch Zone* (viz Obr. 24) Indikaci pozice označuje šedivý vyplněný kruh. Čím blíže jsme k dotykové zóně, tím se kruh zmenšuje. Jakmile se dostaneme do dotykové zóny, kruh zezelená, což indikuje kliknutí myši. Pokud jsme například ve webovém prohlížeči, je možné po tomto zbarvení tahem nahor, dolů nebo do stran nahradit scrollování kolečkem. Pokud vydržíme na místě, objeví se okolo zeleného indikátoru čtverec, signalizující pravý klik myši.

Your interaction space is split into two zones:
the hover zone, and the touch zone.



The hover zone is used for aiming and the touch zone is used for creating touch events on the screen.

Obrázek 24: Touchless for Windows[22]

Stejně jako nám pohyb umožňuje scrollovat, umožňuje nám i tzv. drag and drop¹⁵ a to nejenom s objekty na ploše, ale i s objekty v jednotlivých aplikacích, proto není problém ovládat tímto softwarem aplikace pro Aktiveboard. Aplikace podporuje i několik monitorů a má možnou nastavitelnost ovládání tzv. *základní* a *pokročilou*.

	Number Of Fingers		
Modes	1 Finger	2 Fingers	3+ Fingers
Basic Mode	Click	Scroll	
Advanced Mode	Click , Scroll , Drag and Draw	Zoom , Rotate and Draw	Draw

Obrázek 25: Touchless for Windows[22]

Tento ovládací software je velice jednoduchý a intuitivní. Nemusí se nic dlouze nastavovat. Při zapnutí se automaticky zapne i virtuální klávesnice, která je skrytá na liště nástrojů. Pokud se přepneme např. do MS Word, klávesnice se sama objeví. Samozřejmostí je i průvodce, který nás provede začátky používání programu.

Nevýhodu vydím v občasném "ujždění" indikátoru pozice prstu, stejně jako v indikaci více prstů i přes zavřenou ruku. Ale i přes tyto nedostatky se s ovládacím softwarem pracovalo dobře a jednoduše.

¹⁵Drag and drop = táhni a pusť

3.3.2 Airinput Trial



Obrázek 26: Airinput Trial[23]

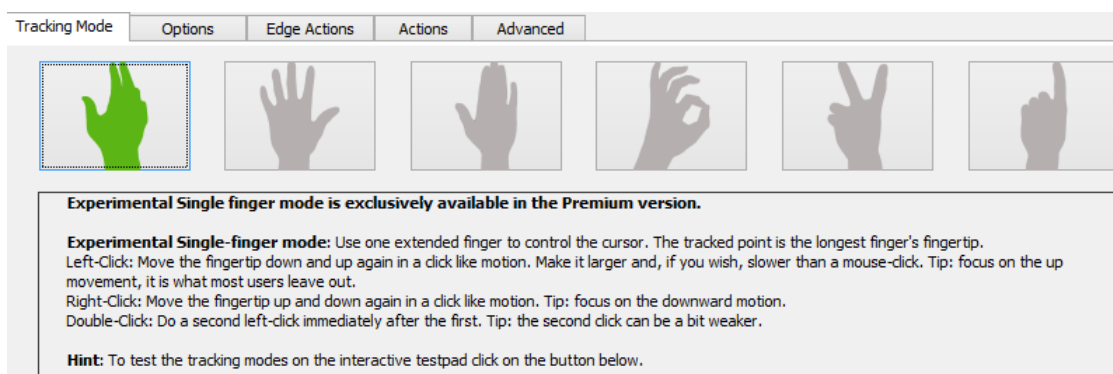
Tento software narozdíl od předchozího nerozděluje snímací prostor, ale ovládá kurzor myši, jež můžeme ovládat přiřazenými gesty podle výběru ovládání (viz Obr. 27). Kurzor je na pozici středu dlaně. Pohyby by měly být plynulé a klidné. Můžeme používat:

- prstový režim
 - Klik levým tlačítkem myši - simulace pomocí pohybu palce ke straně ukazováčku
 - Drag and drop - vydržení v předchozí pozici a tažení na požadované místo
 - Pravý klik - palec a malíček k sobě
 - Dvojklik - rychlé zopakování kliku levým tlačítkem myši
- režim dlaně - Umožňuje drag and drop
- režim úhlu - ovládání pomocí otáčení dlaně kolem středové osy

Program ve verzi *premium* disponuje dalšími třemi způsoby ovládání:

- sevření do "O" - experimentální ovládání
- režim nůžek - experimentální ovládání pro levý stisk a drag and drop
- ovládání jedním prstem

Další nastavení se týká velikosti ohraničení, které omezuje kam až bude možné sledovat pohyb ruky či prstů a citlivost pohybu a gest. Pod záložkou s názvem *Edge Action* se nachází nastavení pro akce, pokud přesuneme okno, nebo objekt do určitého pozice na okraji monitoru. Tyto oblasti se nechají zobrazit a upravit.



Obrázek 27: Airinput Trial

Tato ovládací aplikace je přívětivá, gesta v prstovém režimu jsou intuitivní. Citlivost v základním nastavení bohatě dostačuje k plynulému ovládní. K "ujždění" kurzoru dochází ojediněle. Nevýhodou je občasné přestání fungování při zavření okna, to je možná způsobeno časově omezenou verzí. Ta dovoluje aplikaci spustit několikrát a ovládat počítač 10 minut denně.

3.3.3 Pointable

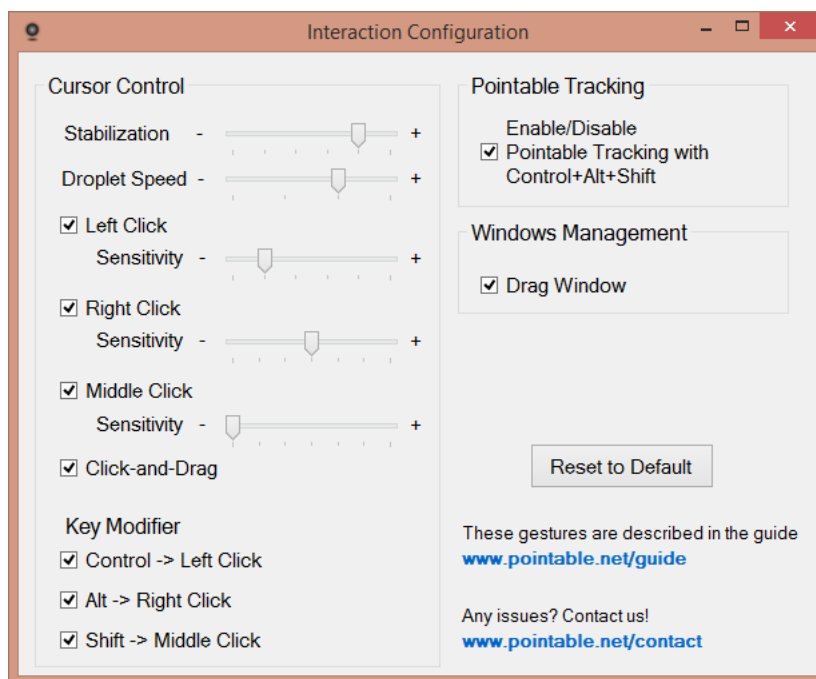


Obrázek 28: Pointable[21]

Software pro ovládní desktopových aplikací. Cena €4,59. Software umožňuje klik, dvojklik, klik pravým i prostředním tlačítkem, přesouvat okna, zavírat okna a to vše jednou rukou. Pomocí kláves SHIFT, ALT a CTRL je možné přizpůsobit kliknutí. Toto se náchá v nastavení vypnout. Stejně jako se nechají nastavit citlivost, rychlost dropletu (zelené kolečko, které se ukáže na místě, kde chceme kliknout). Pomocí rotace prstu ve směru či protisměru hodinových ručiček nám dovoluje scrollování na stránce či ve složce.

- klik levým tlačítkem - dojet prstem na pozici a přejet nahoru a dolů přes zelený kroužek

- klik pravým tlačítkem - dojet prstem na pozici a přejet doprava a doleva k zelenému kroužku
- klik prostředním tlačítkem - dojet prstem na pozici a přejet doleva a zpět k zelenému kroužku
- SHIFT - simuluje prostřední tlačítko myši
- ALT - simuluje pravé tlačítko
- CTRL - simuluje levé tlačítko
- roztažení prstů - vyvolá nabídku programu
- ruka s roztaženými prsty - vyvolá funkci přepínání oken (alt + tab), rotací dlaně se mění okna
- rotace prstem - skrolování
- roztažení ukazováčku a palce nad oknem - aktivace windows managementu (viz níže)
- návod - <http://pointable.net/guide/>



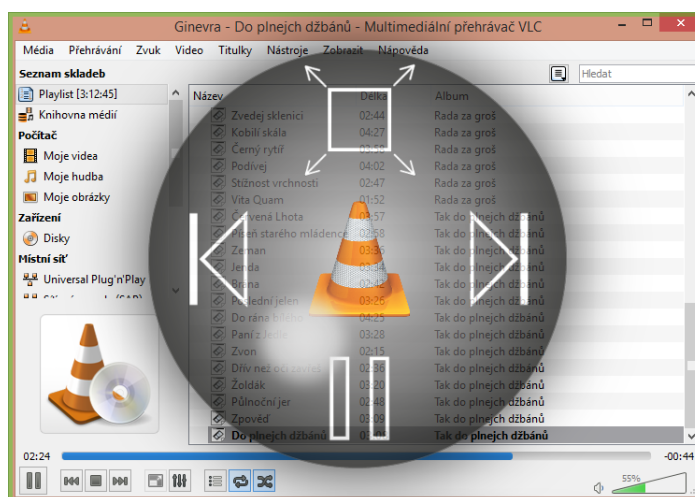
Obrázek 29: Pointable - nastavení

Program má zabudován tzv. windows management. Což je funkce, která umožňuje pozicovat okna a to na polovinu, čtvrtinu, minimalizovat, maximalizovat, nebo vypnout.



Obrázek 30: Pointable - pozicování oken

Rožtažením prstů se objeví nová, kruhová nabídka tzv. programová. Pokud dlaní dojedeme vždy daným směrem (nahoru, dolů, doprava, doleva) zachytíme danou ikonu a přesunem na střed ji použijeme. Některé programy umožňují i rotaci dlaní. Takto můžeme například měnit hlasitost ve Windows Media Playeru.



Obrázek 31: Pointable - nabídka programu

Více k tomuto softwaru můžeme nalézt v online průvodci. Tento program nabízí celkem širokou škálu možností a je jen o zvyku se ho naučit používat. Jeho největší slabinou je ale rychlá detekce pohybu prstu, takže snímá i mírné třesy. To vytváří problém udržet droplet na místě.

3.3.4 Mudra mouse

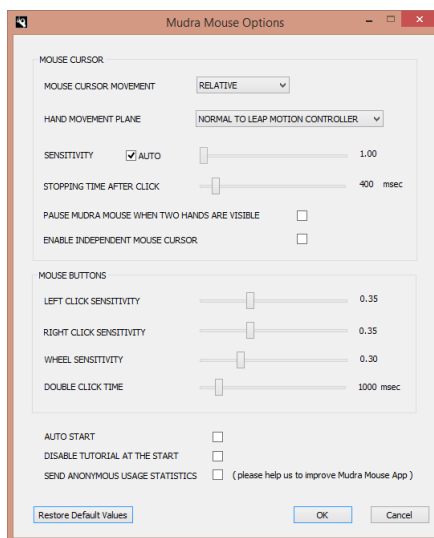


Obrázek 32: Mudra mouse[31]

Aplikace je dostupná pro MS Windows i MacOS. Jedná se o jednoduchou, intuitivní a nenáročnou aplikaci, která simuluje:

- levý klik - palec a ukázováček k sobě a od sebe
- drag and drop - palec a ukazváček k sobě, držet a táhnout
- pravý klik - palec a prostředníček k sobě a odsebe
- scrollování - krouživý pohyb prstem

Aplikace disponuje jednoduchým nastavením citlivosti, odečítání pozice kursoru, časových prodlev, rychlost scrollování, atd.



Obrázek 33: Mudra mouse - nastavení[31]

3.4 Shrnutí

Všechny tři výše zmíněné aplikace mají potenciál ovládat počítač a programy, které nejsou určené pro Leap Motion. Při vydání technologie Leap Motion pro veřejnost byly dostupné právě Touchless ve své základní verzi. Tehdy bylo jeho ovládání trochu těžkopádné. To se ale změnilo s příchodem nové verze softwaru pro kontroler.

Program Pointable má své počátky také v začátcích vydání Leap Motion, nicméně nastavit ho co nejlépe je umění pro mistry. Program Airinput a Mudra mouse jsou nejnovější. Věřím tomu, že Airinput ve své *premiimu* verzi, je druhý nejvýhodnější kandidát, hned po Mudra Mous, pro ovládání aplikací pro interaktivní tabule a systém. Z toho tedy vychází i vyhodnocení aplikací. Kritérii byla jednoduchost, intuitivnost, jednoduché nastavení, stabilita pohybu.

1. Mudra Mouse
2. Airinput
3. Touchless
4. Pointable

Samozřejmě existuje další ovládací software. Například Game WAVE[30] je software pro kontroler, který umožňuje kompletní ovládání počítačových her a systému pomocí různých kombinací gest jednotlivé ruky a to tím stylem, že gestu je přiřazena akce klávesy.

3.5 Zájem učitelů o technologii Leap Motion

Byl proveden dotazníkový průzkum (viz 8.2) a to v rámci akce "Dalšího vzdělávání učitelů"¹⁶, pořádané katedrou informatiky Jihočeské univerzity v Českých Budějovicích. Vzhledem k nutnosti vyzkoušení aplikací a jejich ovládání nebylo možné udělat průzkum pouze přes video. Dotaník se skládal celkem ze 13 otázek, z čehož pro výzkum byly nejdůležitější:

- Oslovila vás představená technologie Leap Motion?
- Viděli byste potenciál ve využití Leap Motion ve výuce?

¹⁶<http://wvc.pf.jcu.cz/ki/?article=/uchazec/dalsi-vzdelavani-ucitelu-informatiky.html>

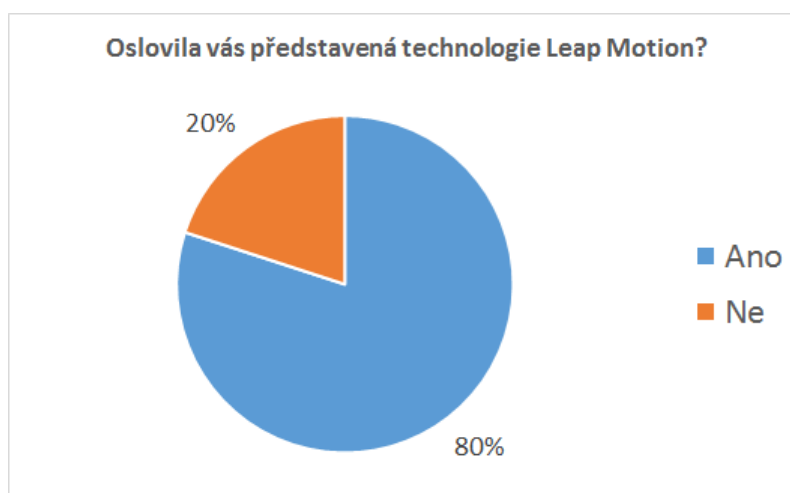
- Co byste u této technologie ocenili, nebo naopak zkritizovali?
- Líbila se vám nabídka aplikací?
- Přišlo vám ovládání vyzkoušených aplikací intuitivní?
- Použili byste zařízení Leap Motion ve své výuce?

Kromě třetí otázky, která byla tvořena dlouhou odpovědí, dávaly zbylé na výběr z možností ano a ne. Jednalo se o učitele technických i přírodovědných oborů různého věkového uskupení s různou dobou učitelské praxe.

3.5.1 Vyhodnocení dotazníku

Oslovila vás představená technologie Leap Motion?

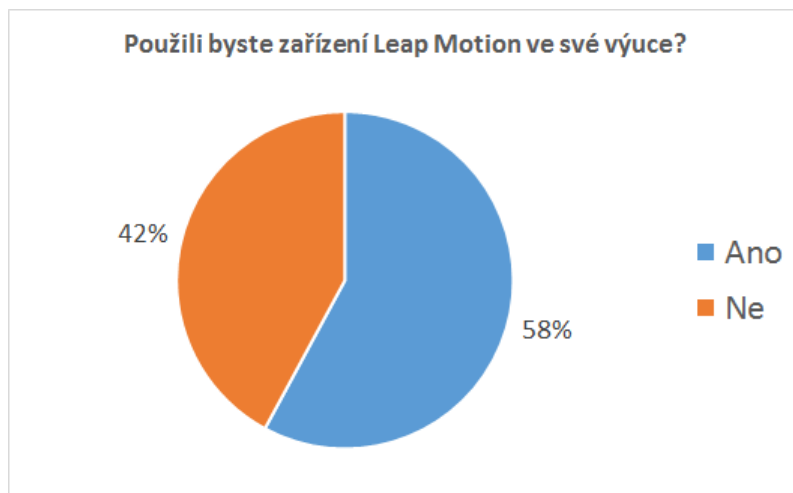
Z celkového počtu 20 dotázaných odpovědělo kladně 16 respondentů.



Obrázek 34: Oslovila vás představená technologie Leap Motion?

Viděli byste potenciál ve využití Leap Motion ve výuce?

Z celkového počtu 20 dotázaných odpovědělo 19, kladně pak 11 respondentů.



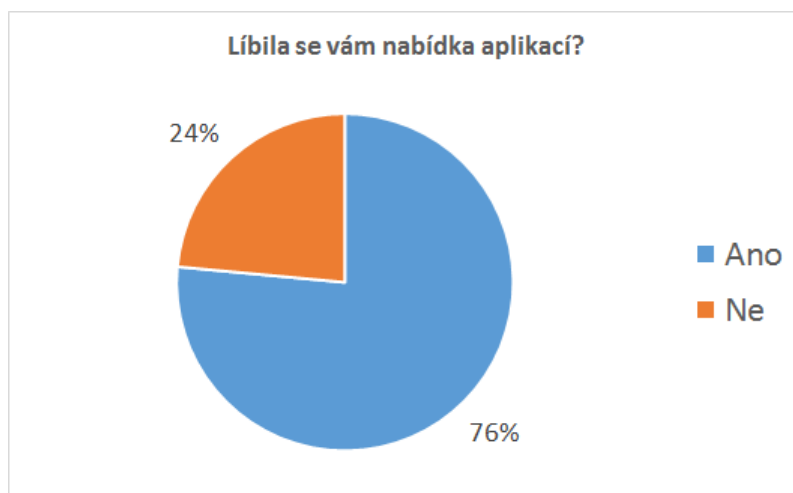
Obrázek 35: Z celkového počtu 20 dotázaných odpovědělo 19, kladně pak 11 respondentů.

Co byste u této technologie ocenili, nebo naopak zkritizovali?

Z celkového počtu 20 dotázaných odpovědělo 14. Většinu se líbila interaktivita, ale odrazovala je složitost ovládání aplikací. Některým vadil i nedostatek softwarového zázemí.

Líbila se vám nabídka aplikací?

Na tuto otázku odpovědělo 17 respondentů, kladně pak 13 respondentů.



Obrázek 36: Líbila se vám nabídka aplikací?

Přišlo vám ovládání vyzkoušených aplikací intuitivní?

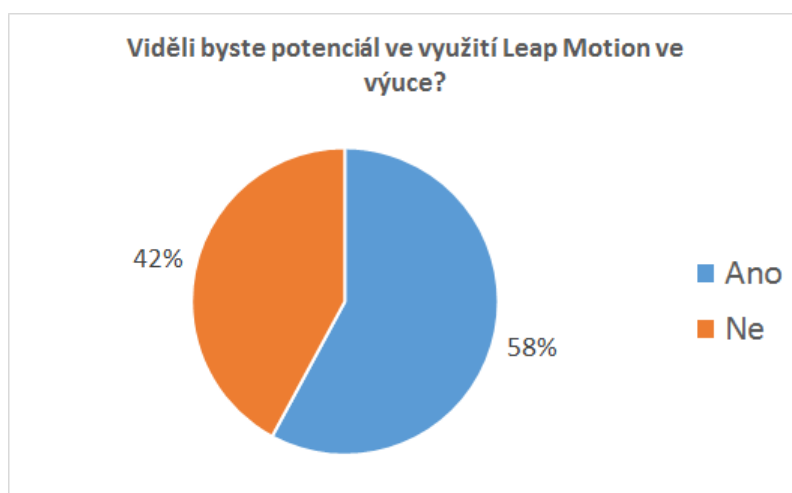
Na tuto otázku odpovědělo 19, kladně pak 14 respondentů.



Obrázek 37: Přišlo vám ovládání vyzkoušených aplikací intuitivní?

Použili byste zařízení Leap Motion ve své výuce?

Na tuto otázku odpovědělo 20 respondentů, kladně pak 12 respondentů.



Obrázek 38: Použili byste zařízení Leap Motion ve své výuce?

3.5.2 Shrnutí dotazníku

Učitele technologie Leap Motion zaujala. Nicméně je odradila nutnost učit se nové ovládání i když jim přišlo intuitivní. To bylo největší zdržení ve výuce. Proto by ho většina učitelů doporučila spíše jako interaktivní doplněk. 60% učitelů by pak tuto technologii použilo ve své vlastní výuce, 55% pak vidí v novém zařízení větší potenciál. Další omezujícím faktorem byl nedostatek výukových aplikací. Kompletní vyhodnocení za pomoci Google Forms v příloze 8.3.

4 Praktická část - vlastní výuková aplikace

V této části se budu věnovat popisu vývoje aplikace pro technologie Leap Motion a to za pomoci použití programů Blender¹⁷, herního enginu Unity 3D a v Unity integrované vývojové prostředí MonoDevelop. To slouží pro psaní skriptů pro jazyk C# a JavaScript. Disponuje jako většina IDE¹⁸ našeptávačem bohužel bez nápovědy.

4.1 Téma aplikace

Zvolené téma se jmenuje Sluneční soustava. To jsem vybral proto, že mě vesmír vždy fascinoval a proletět se po mléčné dráze a podívat se na planety z blízka, jako z nějakého dokumentárního filmu by mohlo žáky taktéž velice zaujmout. Důležitá stránka aplikace je i její vizualizace jako textury, sluneční záře, asteroidové pásy, rotace planet a jejich měsíců, popřípadě nějakého vesmírného satelitu, např. ISS¹⁹

Bohužel jsem s 3D grafikou nikdy nepracoval a tak bylo obtížné dohnat alespoň minimum potřebných znalostí. Stejně tak bylo nutné projít spoustu tutoriálů[22] k Unity a využívat všemožné zdroje, aby bylo docíleno chtěného výsledku.

4.2 Popis aplikace a její cíle

Cílem aplikace je ukázat žákům co nejvěrněji a nejjednodušeji naši Sluneční soustavu. Ovládání pomocí kontroleru by mělo být intuitivní a co nejjednodušší. Informace k planetám a měsícům jsou přechvzány ze stránek Aldebaran[44] a informačního serveru NASA[45]. Aplikace se skládá ze čtyř hlavních scén:

- Intro
- Hlavní scéna
- Scéna porovnání planet
- Kvíz

¹⁷<http://www.blender.org/>

¹⁸Integrated development enviroment

¹⁹International Space Station - mezinárodní vesmírná stanice

Intro je vlastně scéna s menu. Zde bylo vytvořeno několik tlačítek a to:

- Sluneční soustava - přepne do hlavní scény
- Porovnání objektů
- Kvíz
- O aplikaci - informace o autorovi, a zdrojích použitých k vytvoření aplikace
- Konec - ukončí aplikaci

Hlavní scéna pak obsahuje klíčové prvky aplikace. Jedná se o mode Sluneční soustavy, který je možno "prolétnout" díky kontroleru Leap Motion. Ten obsahuje jednotlivé planety a pás asteroidů mezi Marsem a Jupiterem. Poslední scéna je porovnání planet v měřítku vzhledem ke Slunci.

4.3 Vývoj aplikace

Vývoj aplikace procházel několika fázemi.

1. V první fázi byla navrhnutá myšlenková mapa (viz Obrázek 55) s tím, co vše by měla aplikace obsahovat. Rozvrhnutí a popis scén, navržení ovládání, navrhnutí uživatelského rozhraní.
2. Tato fáze byla zaměřena na nabytí znalostí co se týče 3D grafiky a používání Unity 3D [22]. Tato část zabrala nejvíce času a dá se říci, že mě provázela i zbylými fázemi, protože se pořád objevovaly nové problémy.
3. V této části vývoje byly vytvořeny jednotlivé scény. byly propojeny, psány skripty, vkládány modely.
4. Před konečnou finalizací bylo nutné aplikaci otestovat alespoň v možných dostupných podmínkách. Toto testování probíhalo na spolužácích, ve škole a v rámci rodiny.
5. Konečná fáze vývoje aplikace. Zde došlo na skompletování aplikace.

4.3.1 Prostředí Unity 3D

Prostředí Unity 3D není software pro modelování 3D grafiky, jak by se mohlo z názvu zdát, ale jedná se o herní engine. To znamená, že pracuje s již hotovými modely a objekty, na které přidává jednotlivé komponenty, které zajišťují objektu různé funkce, jako fyzikální zákony, neprůchodnost světelných paprsků, textury, kolize, navigace, hudební zdroje, skripty a mnoho dalšího.

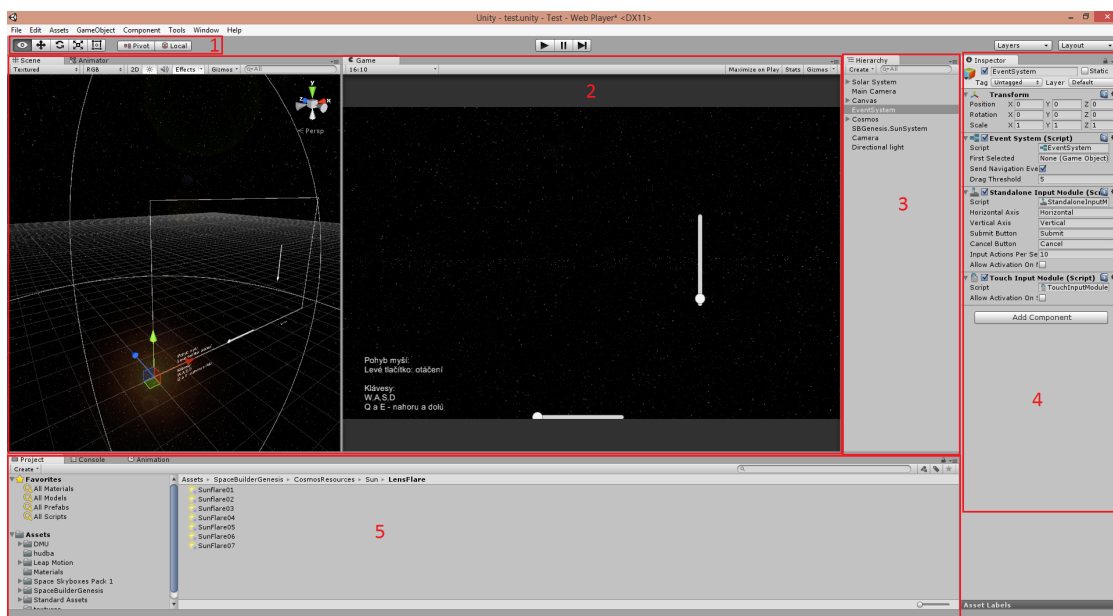
Rozvržení okna je není nic neobvyklého. S podobným rozdělením komponent se setkáme ve většině 3D editorů. navíc se nechají jednotlivé komponenty přesouvat tak, jak to vyhovuje uživateli.

1. Nástroje pro základní editaci objektů (posuv, rotace, změna velikosti, ...)
2. Hlavní okno editoru - v mém případě rozdělené na rendrovací část a část scény
3. Záložka se seznamem objektů vložených ve scéně
4. Inspektor - Zde se nastavují veškeré vlastnosti vybraného objektu a přidávají komponenty
5. Lišta s několika záložkami
 - (a) Project - složky projektu, Unity pracuje s tzv. Assets (jednotlivé složky se vším, co potřebujeme ke tvorbě)
 - (b) Console - klasická console pro hlášení chyb
 - (c) Animation - okno animátoru, kde je možné klíčovat jednotlivé snímky a animovat pohyb a chování objektů

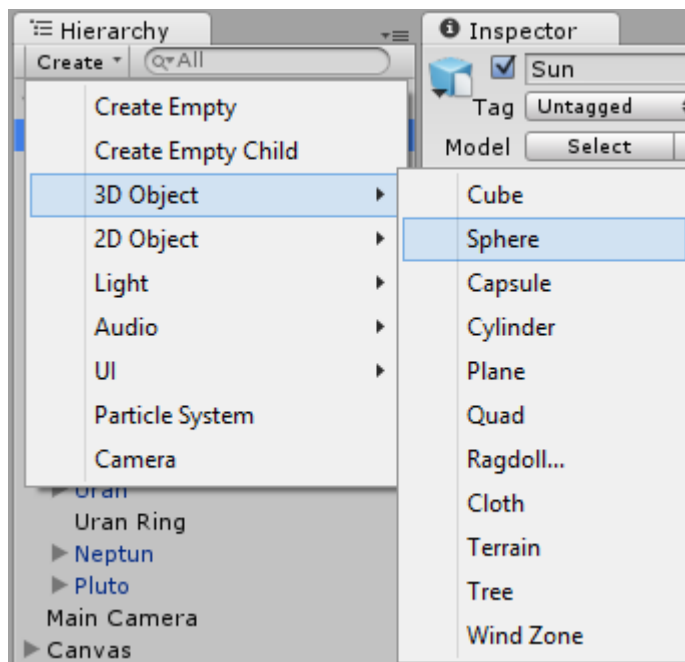
Pohyb v okně scény je pomocí myši. Prvé tlačítko posun scény, scrollováním zoomujeme a pravým se provádí rotace scény pro lepší náhled a přístup k objektům. Díky oknu projektu můžeme textury, skripty, shadery, objekty a další věci přiřazovat objektům napřímo pouhým presunem na určené políčko v Inspektoru.

I přesto, že s enejdná o modelovací program, dokáže Unity vytvořit pár základních 3D objektů. Bohužel už je nedokáže modifikovat na vyšší úrovni než je zmenšování, zvětšování, posun, nebo rotace. To sice omezuje, ale od toho je zde software třetích stran jako Blender nebo Daz 3D²⁰.

²⁰<http://www.daz3d.com/home>



Obrázek 39: Okno Unity editoru

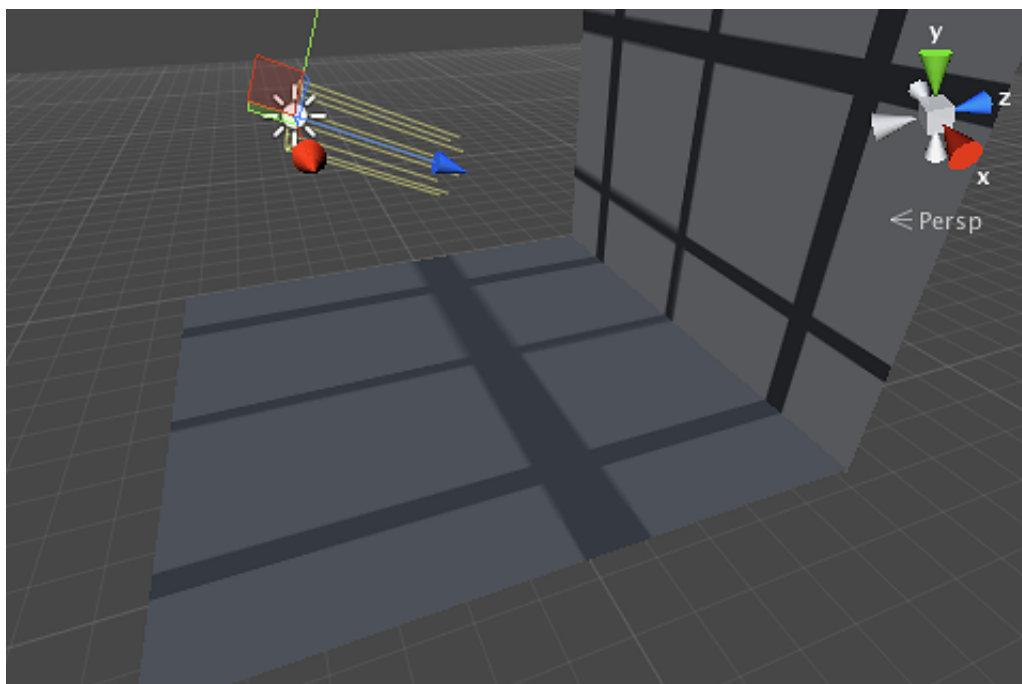


Obrázek 40: Okno Unity editoru - tvorba základních 3D objektů

Dalšími potřebnými prvky pro práci na scéně je zvukové pozadí, částicový systém a hlavně práce se světlem[23] a to:

- všesměrové světlo - nechá se nastavit radius
- bodové světlo - nechá se nastavit poloměr dosahu a úhel osvětlení (světlomet)
- směrové světlo - svítí jedním směrem

Bohužel prostředí Unity ve verzi *zdarma* neumí vrhat stíny při použití všesměrového světla. Což je problém, pokud chceme znázornit stín an pohybující se předmětu. Naneštěstí se tento problém nechá obejít použitím bodového světla, které sleduje pohybující se předmět. Některá světla mohou využít tzv. *cucoloris* nebo *cookies*. Což je vlastně maska, která umožňuje vytvářet různé druhy stínů [24].



Obrázek 41: Směrové světlo - Cucoloris

Detailnější popis všech funkcí a práce s nimi je samozřejmě dostupný v online dokumentaci [25].

Ještě je potřeba se více zmínit o tzv. assetech. Což jsou předpřipravené scény, skripty, generátory, které usnadňují vývoj aplikace. Tyto assety lze získat z tzv. asset store [26], což je obchod právě pro tyto předpřipravené dá se říci doplňky pro Unity 3D. I já sám jsem jeden využil a to SpaceBuilder: Genesis [27]. Ten umožňuje vytvoření vesmírného

prostoru i s tvorbou planet. Tento doplněk byl ale v mé práci využit pouze k vygenerování hvězdného pozadí a pásu asteroidů mezi planetou Mars a Jupiter.

Některé assety jsou zdarma. Jsou rozděleny do různých kategorií podle potřeby uživatele. Jediné omezení bývá použití na jednotlivé verze Unity, konkrétně na verzi *Pro*. Dále je zde možné získat i zvukové soubory, kompletní projekty, skripty, částicové systémy, 3D modely, animace, textury a materiál, shadery²¹ a další.

4.3.1.1 Skriptování

Unity 3D umožňuje skriptovat ve dvou jazycích:

- C#
- JavaScript

Každý objekt může mít až několik skriptů. Jak bylo zmíněno výše, Unity disponuje prostředím MonoDevelop. Pokud je proměná ve skriptu jako `public` objeví se v prostředí Unity jako vyplňovací políčko (viz Obrázek 42). Objekt typu `Transform` může zastupovat jakýkoli objekt, který určíme. To umožňuje velkou flexibilitu programu, protože z daného objektu pak můžeme dostávat různé údaje nejen o objektu, ale i komponentách, které objekt má. Každý skript, který se přidává objektu, musí být odvozeny od `MonoBehavior`[36]. To je základní třída, která obsahuje základní metody pro použití v programu. Nejdůležitější z nich je `Update`, která se volá při každém novém framu aplikace.

```
public class MovingCameraByLeap: MonoBehaviour
{
    public Transform target ;
    public float speed = 5.0f;
```

4.3.1.2 Animace

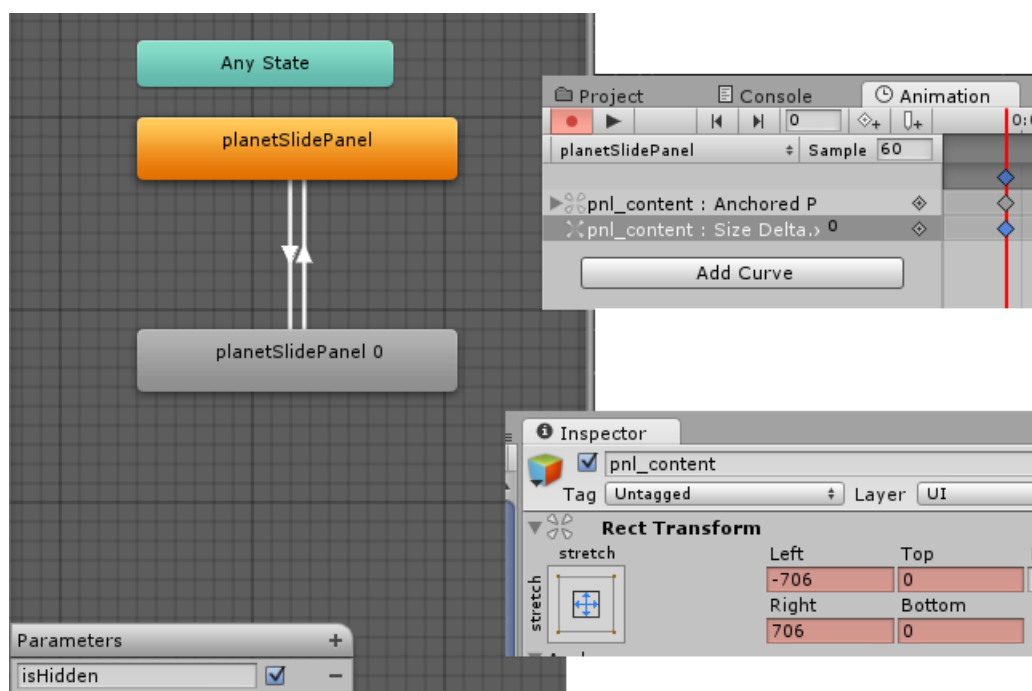
Prostředí Unity disponuje i *Animátorem*. Ten funguje na principu klíčování snímků. Pokud chceme vytvářet novou animaci, musíme nejprve vytvořit nový soubor s koncovkou `*.controller`. Poté můžeme v jednom časovém okamžiku nastavit objektu jedny hodnoty a v druhém časovém okamžiku jiné hodnoty. Pokud animaci spustíme, objekt se plynule

²¹<http://cs.wikipedia.org/wiki/Shader>



Obrázek 42: Vliv veřejných proměnných na zobrazení v Unity

přetransformuje do požadovaného tvaru, umístění, textury, ... Animace se v základním nastavení přehrává dokola. Pokud klikneme na soubor animací můžeme toto v inspektoru vypnout. Dále je možné nastavovat pomocí animátoru podmínky, za jakých se bude animace provádět, například v závislosti na proměnných ve skriptu.



Obrázek 43: Animátor, animace, klíčování

Vše potřebné je popsáno v přehledném online dokumentu *Unity - Scripting API*[35]. Jazyk skriptů se nechá kombinovat, to znamená, že na jednom objektu může být skript psaný jak v C# tak v JavaScriptu. Skripty použité v této aplikaci jsou součástí přílohy.

5 Aplikace - Sluneční soustava

Jak již bylo zmíněno v metodě práce, celá aplikace se skládá ze čtyř scén, které jsou popsány v sekci 5.5. Každá scéna obsahuje 2 kamery:

- Hlavní kamera - ta se pohybuje a zabírá scénu
- UI kamera - kamera na kterou je vykresleno uživatelské rozhraní (User Interface)

Pomocí skriptů můžeme přepočítávat koordinace a to například z 3D scény, kdy jsou jako `Vector3` (x, y, z) přepočítány na souřadnice pro obrazovku `WorldToScreenPoint` (`Vector3 vector`) [35], nebo opačně. V jednom případě je potřeba vektor ve třech směrech, v druhém pak pozice myši na obrazovce.

```
MainCam.WorldToScreenPoint(transform.position);
```

5.1 Interakce Leap Motion

Jak vytvořit nový objekt kontroleru je popsáno v sekci 2.2.1.1. Tato kapitola se zabývá tím, jak je to s ovládáním velikostí planet, rychlosti rotace, a pohybem po scénách a po obrazovce.

5.2 Pohyb po obrazovce

Celá interakce kurzoru stojí na kódu, který vytvořil Pohung Chen [38]. Jen je mírně upravený pro detekci kliknutí (viz 5.3.1.1). Pozice ukazováčku je znožorněna šedivou kružnicí. Čím blíže se ukazováček blíží k monitoru, tím se kružnice zmenšuje, dokud nedosáhneme interakční vzdálenosti, což je vlastně nulová pozice. Jedná se v podstatě o virtuální dotykovou plochu.

Celá detekce se skládá ze dvou tříd `LeapFingerDraw.cs` a `LeapInput.cs`. První vykresluje kruhový kurzor na pozici ukazováčku, druhá zajišťuje, jestli se ukazuje a jak daleko jsme od nulové pozice. Pokud se dostaneme za virtuální dotykovou plochu, změní se kurzor na malé zelené kolečko. Pokud opět odjedeme ze stejné pozice, detekuje se toto jako kliknutí na dané pozici. Vše by mělo být prováděno plynule, bez trhaných pohybů.

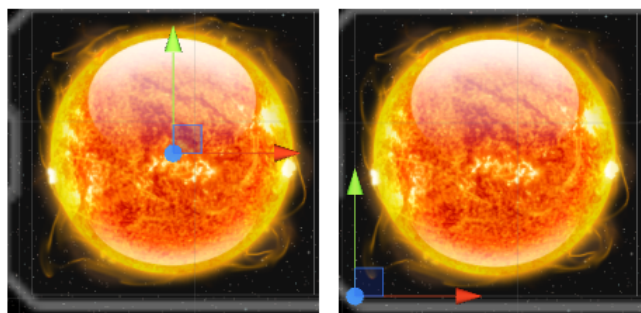
5.3 Pohyb ve scéně

Pohyb ve scéně se skládá ze tří základních akcí:

- Interakce s UI a planetami
- Pohyb po scéně
- Ovládání gesty

5.3.1 Interakce s uživatelským rozhraním a planetami

Část této sekce byla popsána více (viz 5.2). Proto se bude zabývat tlačítky a jejich metodami. Přes okno hierarchie a tlačítko *create* nebo přes lištu nástrojů *GameObject* vložíme UI prvek *Button*. Ten se automaticky vloží do *canvasu*. Každý objekt může být potomkem jiného. Od toho se poté odvíjí hodnota pozice souřadnic. Každému objektu také můžeme nastavit pozici tzv. *pivotu*. To je bod, od kterého se hodnoty přičítají či odčítají.



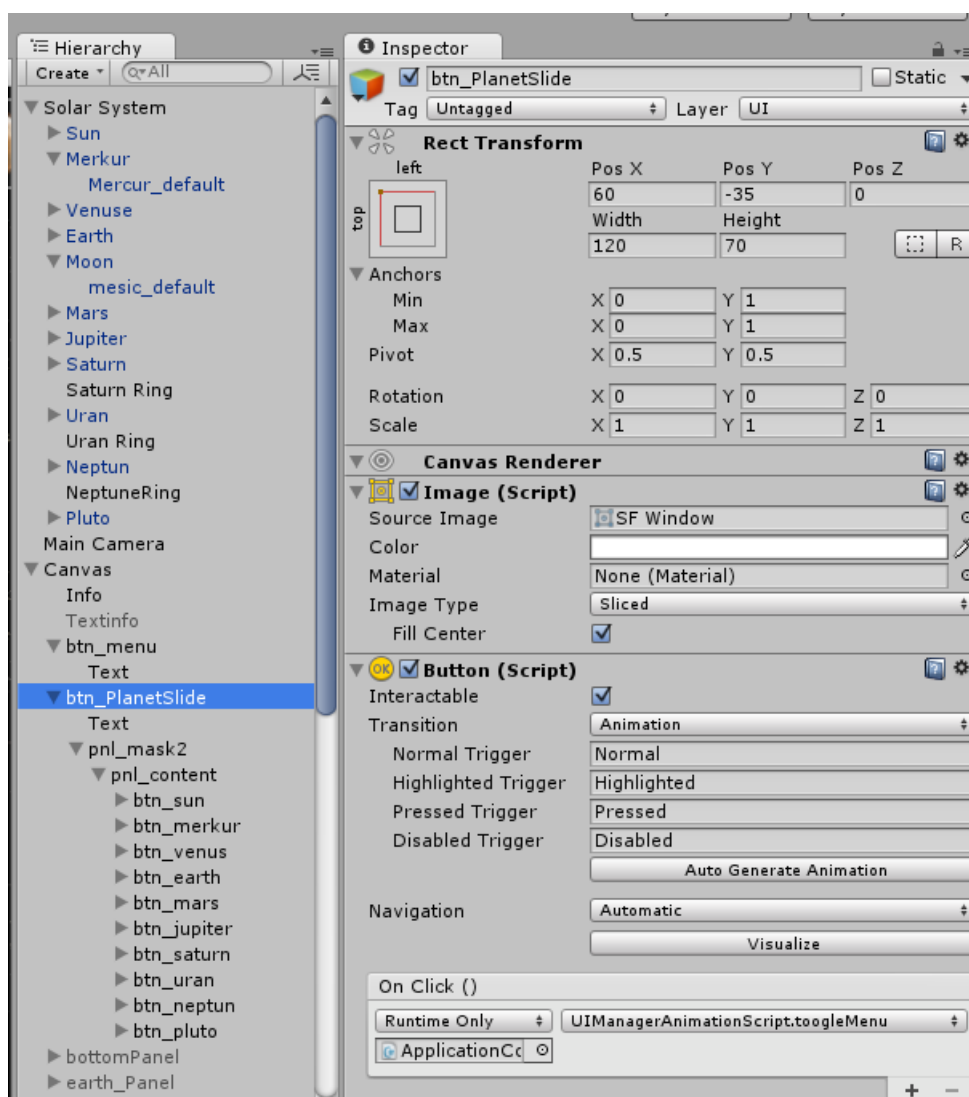
Obrázek 44: Pozice pivotu: na levo (0.5 , 0.5) v pravo (0, 0)

Grafika pro tlačítka planet jsou od Dana Wiersema[41] a pro zbylá tlačítka z volně dostupného setu od Unity[40]. Fonty jsou od Ndiscovered [42] a Tracer Tong[43].

Tlačítko se samo o sobě skládá defaultně z několika komponent a to *Image* a *Button*. Obě komponenty jsou skriptpy. *Image* nám říká, jestli má tlačítko nějaký obrázek, *Button* nám umožňuje nastavení stavových akcí jako *highlighted*, *pressed*, *disabled*, *normal*. Umožňuje několik možností interakce:

- Animace
- Změna barvy
- Změna ikony

Animace umožňuje animovat jednotlivé stavy tlačítka za pomoci animátoru (viz 4.3.1.2). Změna barev znamená jednoduché podbarvení nebo zbarvení tlačítka při daném stavu a v neposlední řadě můžeme měnit obrázek ikony. Skript `Button` také obsahuje přednastavenou funkci `OnClick()`. Přidáním akce se zobrazí položka, která vyžaduje objekt se skriptem, ve kterém se nachází funkce, kterou chceme použít. Po přidání objektu nám rozbalovací menu nabídne přehled komponent. Vybereme námi požadovanou a vybereme konkrétní metody. Aby byla metoda přístupná, musí být *public* a *void*. Jako argument může být použita číselná hodnota nebo herní objekt.



Obrázek 45: Vlastnosti tlačítka

Tlačítko na obrázku má na metodu `OnClick()` přidán objekt `ApplicationController`, který obsahuje komponentu (skript) `UIManagerAnimationScript` a tato třída poté obsahuje metodu `toggleMenu()`. Ta zajišťuje vyjždění panelu s tlačítky s planetami.

```
public Animator contentPanel;

public void toggleMenu(){
    contentPanel.enabled = true;
    bool isHidden = contentPanel.GetBool ("isHidden");
    contentPanel.SetBool ("isHidden", !isHidden);
}
```

5.3.1.1 Kliknutí aneb zobraz panel pomocí Leap Motion

Kliknutí na planetu pomocí Leap Motion je uskutečněn pomocí tzv Ray Castu. Jedná se vlastně o paprsek vyslaný, z kamery na pozici kurzoru, dopředu. Tento paprsek má několik variant. Buď se vrátí po nárazu na první objekt nebo prolítne do nekonečna a vrátí list všech objektů, kterými proletěl. Níže je ukázka kódu pro zjištění kliknutí na planetu. V Unity je nejsnazší hledat objekty pomocí názvu. Ten musí odpovídat názvu v editoru.

```
RaycastHit hit; //vytvoření nového paprsku
Ray rayTest = MainCam.ViewportPointToRay(screenPos); //odkud paprsek
    bude vycházet
if(Physics.Raycast(rayTest, out hit)){ // pokud na něco narazí
    ObjectName = hit.transform.gameObject.name; //zjistí název objektu do
    kterého si narazil
    ObjectClicked oc=GameObject.Find (name).GetComponent<ObjectClicked>();
    //z nalezeného objektu dostaneme komponentu
    oc.SetActive(); //provedená akce (zobrazení infromačního panelu)
}
```

Bohužel UI pracuje trošku jinak. Zde musíme použít `PointerEventData`. Zbytek už je podobný předchozímu případu. Zde dostáváme však seznam objektů.

```
PointerEventData pointer = new PointerEventData(EventSystem.current); //objekt
    eventu ukazatele
pointer.position = UIcam.ViewportToScreenPoint(screenPos); //výchozí pozice
List<RaycastHit> rayResult = new List<RaycastHit>(); //list objektů
EventSystem.current.RaycastAll(pointer, rayResult); //vyslání paprsku
```

```

if(rayResult.Count >0){
    //našli jsme něco?
    buttonName = rayResult[0].gameObject.transform.parent.gameObject.name;
    //najdi nazev objektu
    Button button = GameObject.Find (name).GetComponent<Button> (); //vytvoř
    objekt tlačítka
    button.onClick.Invoke (); //spust' akci na tlačítku}

```

Celý kód je dostupný v příloze (viz příloha: 8.1.1).

5.3.2 Pohyb po scéně

Pohyb po scéně je realizován otevřenou dlaní s roztaženými prsty a to tak že:

- Pohyb blíže k monitoru je pohyb dopředu, naopak dále od monitoru dozadu (posun v ose Z). Čím je vzdálenost od středu kontroleru větší, tím je rychlost větší.
- Pohybem ruky doprava nebo doleva se otáčí kamera v ose Y.
- Rotací dlaně okolo osy se otáčí kamera v ose Z.
- Zvedneme-li dlaň nahoru nebo dolů, rotuje kamera okolo osy X.
- Pohyb dále či blíže ke kontroleru znamená posun kamery nahoru nebo dolů (posun v ose Y). Rychlost se mění se vzdáleností.

Ukázka kódu rotace kamery v pravo a v levo.

```

void rotateRL(Hand hand) {
    float rotThreshold = 0.0f;
    float handX = hand.PalmPosition.ToUnityScaled ().x;

    if (Mathf.Abs (handX) > rotThreshold) {
        transform.Rotate (Vector3.up, handX * 2.0f);
    }

}

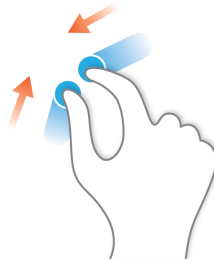
```

Celý kód je k dispozici jako příloha (viz 8.1.2).

5.3.3 Ovládání gesty

Ve scéně se objevují tři základní gesta:

- Rotace ukazováčkem - scale planet
- Rotace ukazováčkem a prostředníčkem - rychlost rotace planet
- Pinch gesto - jemnější posun kamery dopředu nebo dozadu (palec a ukazováček k sobě ⇒ posun ruky ⇒ uvolnění prstů)



Obrázek 46: Pinch gesto

Ukázka níže ukazuje kód pro zvětšování a zmenšování planet. Celý kód je dostupný v příloze (viz 8.1.3)

```
for (int i = 0; i < gl.Count; i++) {
    Gesture gesture = gl[i];
    //velikost planet pomocí kruhového gesta ukazováčku
    if(gesture.Type == Gesture.GestureType.TYPE_CIRCLE && finger.Type() ==
        Finger.FingerType.TYPE_INDEX && finger.IsExtended &&
        fl.Extended().Count == 1) //detekce kruhového gesta a prstu, který ho
        vykonává
    {
        CircleGesture circleG = new CircleGesture(gesture);
        Vector circleDirection = frame.Pointables.Frontmost.Direction; //jakým
            směrem je gesto prováděno
        float angle = circleDirection.AngleTo(circleG.Normal); //úhel
        if(SpaceObjectsScaling.planetScale2 <= 50.0){
```

²¹<http://en.wikipedia.org/wiki/Multi-touch>

```

if(angle<= (3.14/4)&& SpaceObjectsScaling.planetScale2 != 50.0){
    SpaceObjectsScaling.planetScale2 += 1f; //zvětšení velikosti o 1
}else if(angle >= (3.14/4)) //proti směru hodinových ručiček
    double isSmall = SpaceObjectsScaling.planetScale-1.0;
    if(isSmall >= 2.0)
        SpaceObjectsScaling.planetScale2 -= 1f; //zmenšení velikosti o 1
    }
} }

```

5.4 Planety

Pro model planet bylo nutno vyrobit model koule o větším počtu vertexů, protože jinak by při přiblížení byly vidět nepravidelné hrany v daleko větší míře. Toho bylo docíleno v programu Blender 3D. Poté bylo nutno nalézt textury na planety. Ty byly nalezeny na několika webových stránkách [32],[33],[34]. Planety byly v prostředí rozmístěny do vzdálenosti odpovídající v poměru vzdáleností 1000 jednotek od středu což je rovno 1 astronomické jednotce (1 AU). Většina planet má shader typu diffuse nebo bumped diffuse. Rozdíl mezi nimi je ten, že bumped diffuse umožňuje smíchání dvou textur, obyčejné a plastické mapy. Proto textura planety Země vypadá vystoupleji. Některé planety obíhají jejich měsíce. Všechny planety se otáčejí kolem Slunce a kolem své osy.



Obrázek 47: Slunce s panelem a zeleným indikátorem kliknutí

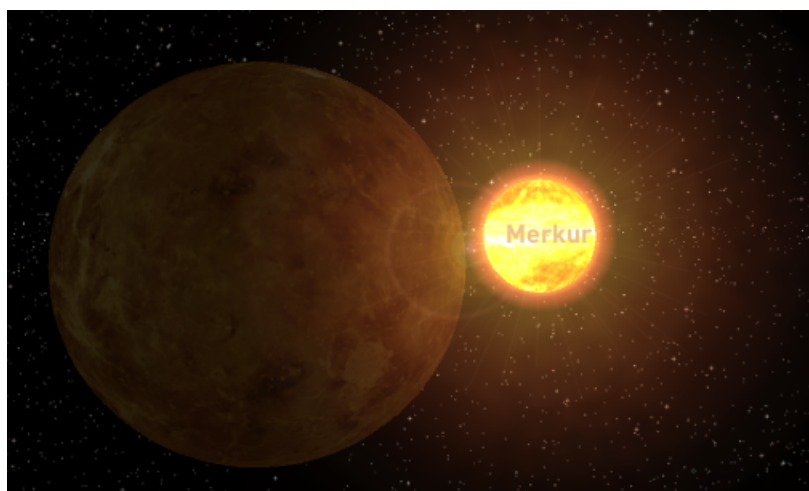
Dále mají všechny planety komponentu `Rigid Body`, která umožňuje ovládání objektu skrz fyziku Unity editor, a `Sphere Collider`, která hlídá kolize a zabraňuje například světelným efektům procházet skrze textury planet. Tyto komponenty jsou potřebné ke zjištění kliknutí pomocí skriptu a metody `OnMouseDown()` a `OnMouseUp()`, které zobrazují informační panel v režimu bez Leap Motion kontroleru[39].

```
public class ObjectClicked: MonoBehaviour{
public GameObject panel;
public Camera cam;
Vector3 startMousePosition;
Vector3 endMousePosition;
void OnMouseDown(){
    startMousePosition = cam.ScreenToWorldPoint (Input.mousePosition);}
void OnMouseUp() {
    endMousePosition = cam.ScreenToWorldPoint (Input.mousePosition);
    if(startMousePosition == endMousePosition)
        panel.SetActive (!panel.activeSelf);}
public void SetActive(){
    panel.SetActive (!panel.activeSelf);}
}
```

5.4.1 Slunce

Jediná planeta s jiným shaderem je Slunce. To má `self-illumin` shader pro průchod světla zevnitř skrze texturu. Zároveň s tím má na sobě několik efektů:

- Všesměrové světlo
- Halo efekt - vytváří jednoduchou iluzi korony
- Rotace textury - simuluje efekt pohybu atmosféry
- Flare - vytváří efekt paprsků



Obrázek 48: Pohled na Slunce z Venuše

5.5 Scény

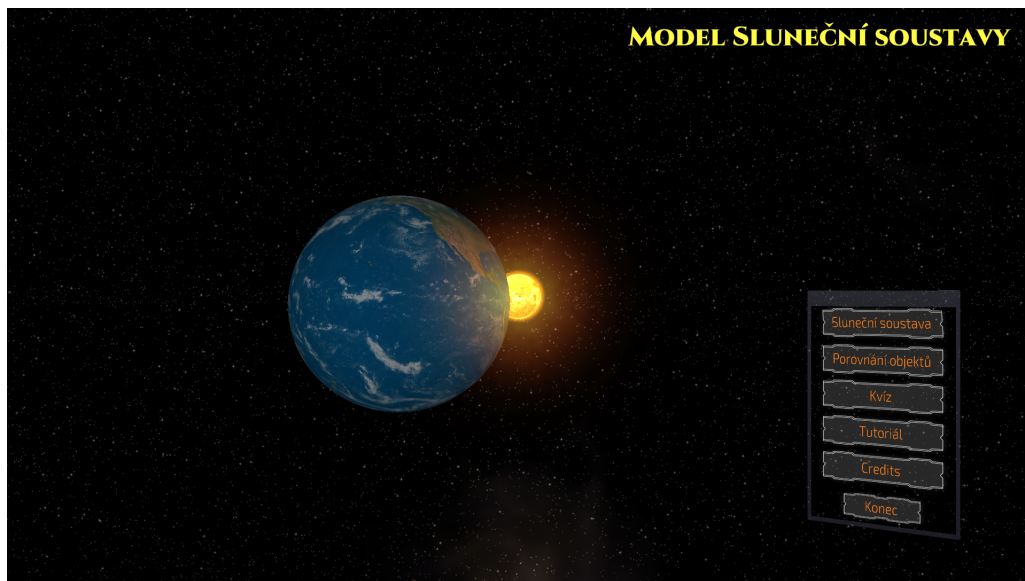
Celá aplikace je rozdělena do čtyř scén a to:

1. Intro - úvodní obrazovka s menu
2. Hlavní scéna - stěžejní scéna aplikace
3. Scéna porovnání - scéna pro porovnání velikosti planet a hvězd
4. Kvíz - jednoduchý kvíz složený z infomací v aplikaci

5.5.1 Intro

Intro je úvodní obrazovka. Obsahuje vlastně jenom rozcestník ro jednotlivých scén, titulky, tlačítko pro ukončení aplikace a tutoriál. Pokud se během několika sekund nepřepneme do jiné scény, tak se změní pohled kamery a to tak, že dojde k Zemi a natočí se na slunce. Titulky s poděkováním a zdroji se zobrazí v posuvném panelu. Tutorial je panel s obrázky pohybu ruky.

Ukázka kódu načítání scény. Jednotlivé scény musí být přidány do build dialogu (viz Obrázek: 53). Ve scriptu se pak scéna načítá buď celočíselnou hodnotou, která je vidět i ve zmíněném dialogu, či textového řetězce jako název scény.



Obrázek 49: Obrazovka intra

```
public void loadMainScene(){
    Application.LoadLevel("mainScreen");
}
public void loadComparationScene(){
    Application.LoadLevel("compare");
}
public void loadMenu(){
    Application.LoadLevel("intro");
}

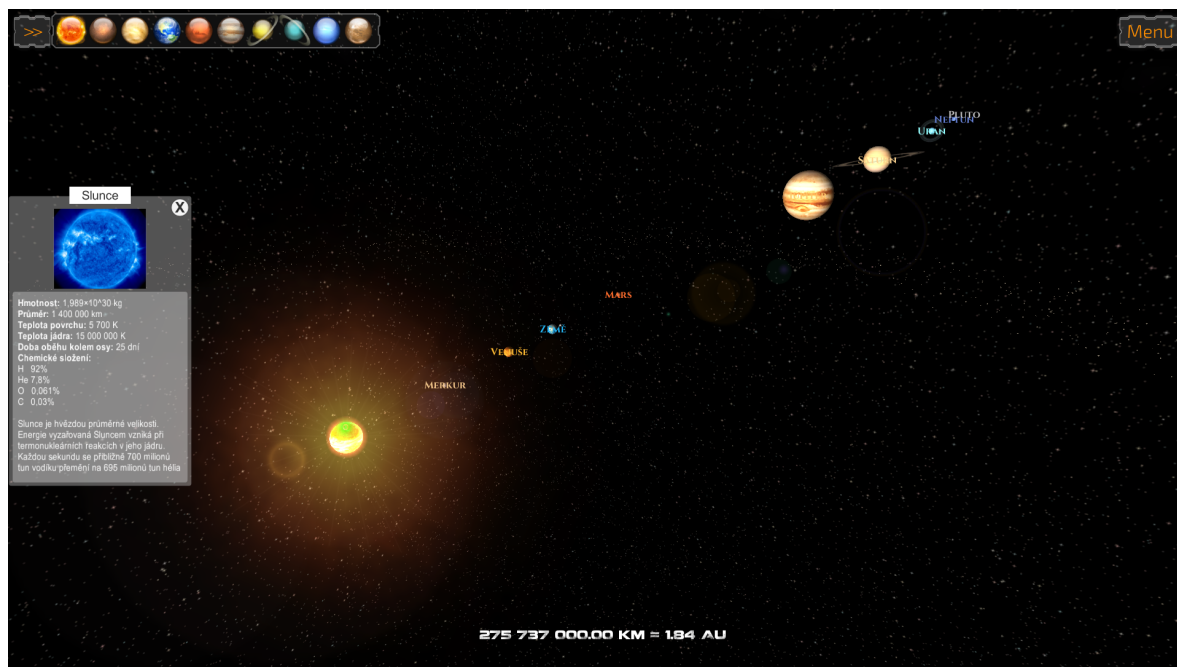
public void exitApp(){
    Application.Quit ();
}
```

5.5.2 Hlavní scéna

Scéna s modelem sluneční soustavy. Tato scéna obsahuje:

- Slunce
- 8 planet
- 1x trpasličí planeta
- Pás asteroidů mezi Marsem a Jupiterem
- Několik měsíců

V pravém horním okraji se nachází tlačítko pro rozbalovací menu. To obsahuje ikony planet. Po kliknutí se kamera přesune k dané planetě a otočí se na Slunce. V levém horním rohu je tlačítko pro návrat do menu. Ve spodní části obrazovky se nachází informační text o vzdálenosti od Slunce. Každá planeta má nad sebou svůj název. Ten zůstává přichycen u boku obrazovky, pokud je planeta mimo přímý dohled. Navíc napomáhá orientaci v tmavém volném 3D protoru. Ovládání scény je popsáno v kapitole 5.2.



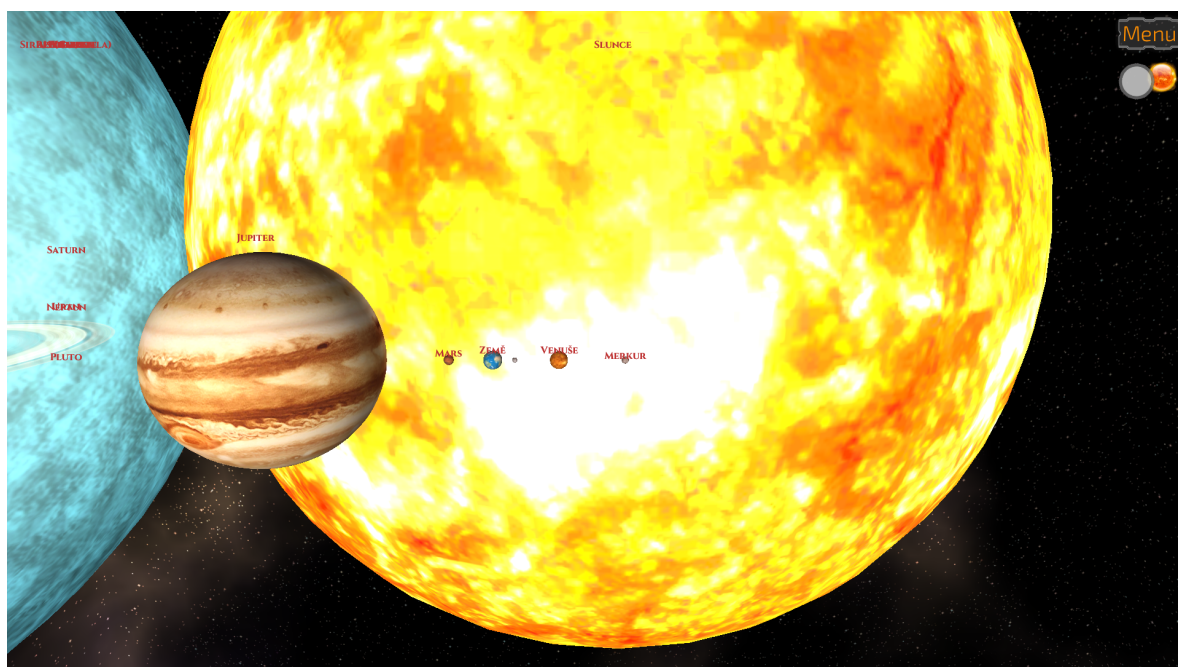
Obrázek 50: Pohled na sluneční soustavu se zvětšenými planetami

5.5.3 Scéna porovnání

Kamera v této scéně začíná na pozici, kdy zabírá Slunce a před ním jsou naskládány planety. Směrem doleva se nachází výběr známých hvězdy větších než naše Slunce a to v tomto pořadí:

1. Slunce
2. Sirius
3. Pollux
4. Aldebaran
5. Rigel
6. Betelgueze
7. UY Scuti

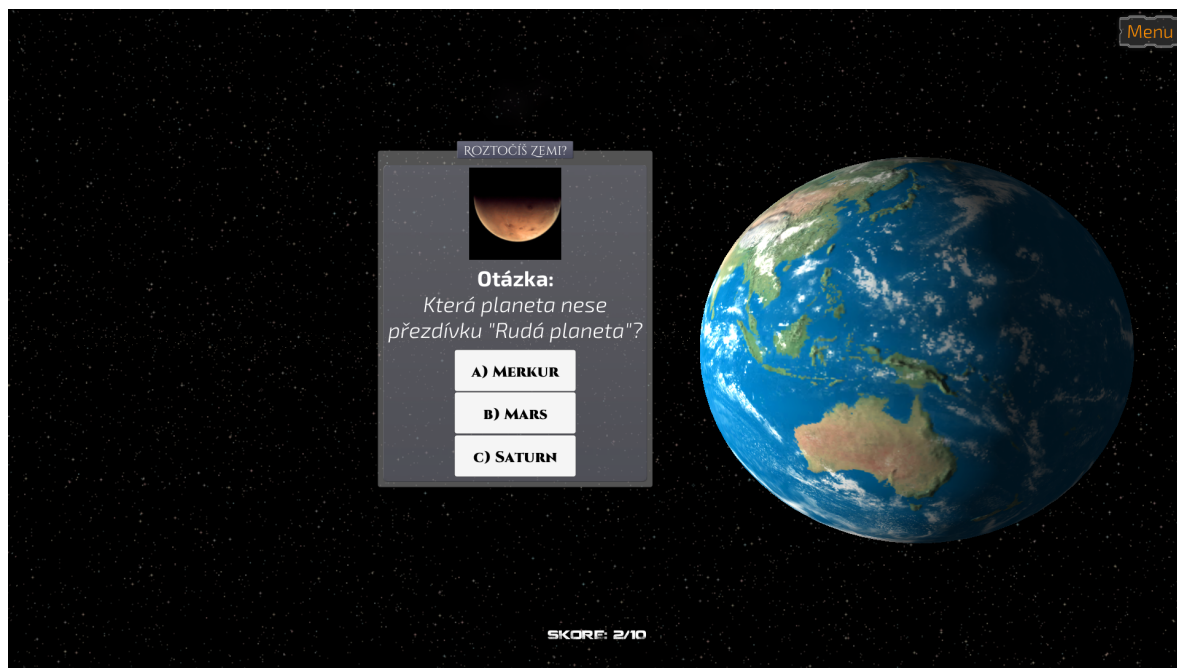
Scéna obsahuje tlačítko pro reset kamery do výchozí polohy. To má ikonu slunce a nachází se pod tlačítkem menu.



Obrázek 51: Porovnávací scéna

5.6 Kvíz

Kvíz je scéna, ve které se snažíte roztočit Zemi počtem správných odpovědí. Ty vidíte pod oknem s otázkou. Odpovědi na jednotlivé otázky lze nalézt v aplikaci po přečtení informací k jednotlivým planetám či měsícům.



Obrázek 52: Porovnávací scéna

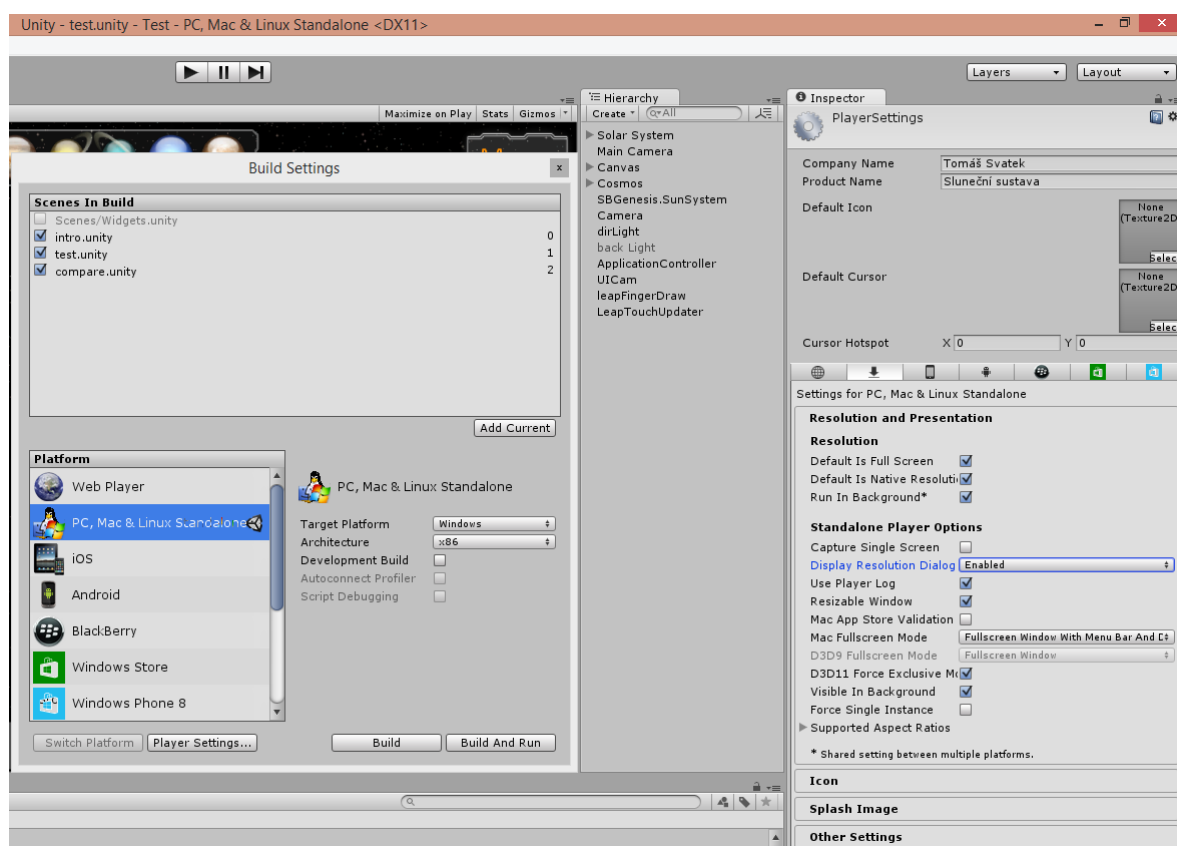
5.7 Zvukový doprovod

V aplikaci je použita hudba z volně dostupných zdrojů z asset storu, a to Epic Orchestral Free Music Pack od Richarda Daskase[46] a Stars a Euphorium Suite od Stephena Lu[47],[48].

5.8 Export a distribuce aplikace

Prostředí Unity 3D je schopno exportovat aplikace do několika formátů a operačních systémů a to od mobilních po ty desktopové. Aplikace je distribuována skrze vlastní webovou prezentaci a to pro Windows x86 a x64, MacOS a Linux. Aplikace se sestavuje stisknutím tlačítka **File** a **Build Settings**. V novém okně si vybereme platformu. Můžeme dát **Build** a program se nás zeptá na uložení. Zadáme název *.exe souboru a program přebere skripty, assety a použije jen to, co je skutečně ve scéně.

Pokud budeme chtít při spuštění dialog s výběrem kvality grafiky či rozlišení musíme zvolit **Player Settings**. To nám v inspektru zpřístupní zmíněná dodatečná nastavení. Dále nám umožňuje nastavit ikonu, zdali aplikace poběží na pozadí, či bude defaultně nastaven fullscreen. Pokud vybereme webovou platformu, můžeme si zvolit barvu templaty a rozlišení okna. Já nastavuji HD Redy tedy 1280 * 720 pixelů.



Obrázek 53: Dialogové okno pro sestavení aplikace

5.9 Alternativní výstup

Vzhledem k testování aplikace pomocí myši a klávesnice, je možné tuto aplikaci ovládat právě přes zmíněná zařízení. Myš se provádí rotace, klávesami W, A, S, D se provádí pohyb dopředu, dozadu a do stran. Klávesami Q a E potom pohyb nahorů a dolů. Pomocí posuvníků měníme hodnoty velikosti planet, rychlost posunu kamery a rychlost rotace.

Unity 3D umožňuje export i pro webovou prezentaci, takže je aplikace dostupná i pro všechny, kteří nemají Leap Motion kontroler, ale chtěli by si s aplikací hrát aniž by ji museli stahovat. Pro spuštění je nutné mít nainstalovaný Unity3D doplněk do prohlížeče.



Obrázek 54: Změna UI pro ovládání klávesnicí a myši

5.10 Ověření aplikace

Aplikace byla ověřena v hodině v 6. ročníku Základní školy Nerudova v Českých budějovicích pod dohledem pana Mgr. Radovana Mikeše. Video z hodiny je přiloženo na CD v příloze 8.4. I přes počáteční překážku s ovládáním, se děti se zařízením rychle sžili a po malé chvíli jim ovládání nedělalo větší problém a aplikace je bavila.

Dále děti zkoušely i jiné aplikace pro zařízení Leap Motion. Postup byl stejný jako u testované aplikace, nicméně pan učitel by zařízení viděl také jenom jako zábavný doplněk výuky, ne ve výuce natrvalo, což potvrdily i výsledky dotazníkového šetření.

6 Závěr

V práci bylo ověřeno, že již napsané aplikace se nechají ovládat Leap Motion kontrolerem, pokud je nainstalován příslušný software, který je dostupný i zadarmo. Bylo popsáno několik výukových aplikací, jakož i technologie Leap Motion a její možný budoucí vývoj.

Byl zjištěn názor učitelů na tuto technologii, který ukázal, že učitelé mají obavu ze složitosti ovládání aplikací a z nedostatku softwaru pro výuku. Aplikace byla ověřena ve výuce. Zde se, i přes malé nedostatky a prvotní problémy s ovládáním, aplikace líbila a děti zaujal nový systém ovládání. Učitel také ocenil možnosti aplikace a její grafické zpracování.

Práce mi přinesla nový pohled na bezdotykovou technologii. Naučila mě pracovat s herním enginem Unity 3D a vymýšlet nové přístupy k problémům i přes desítky hodin strávených koukáním a pročítáním tutoriálů

7 Seznam použité literatury a zdrojů

- [1] Leap Motion. In: *Wikipedia: the free encyclopedia* San Francisco (CA): Wikimedia Foundation, 2013, 23.03.2015 [cit. 2015-03-28]. Dostupné z: http://en.wikipedia.org/w/index.php?title=Leap_Motion&oldid=653086724
- [2] Engadget. *Leap Motion gesture control technology hands-on* [online]. 2012 [cit. 2015-03-28]. Dostupné z: <http://www.engadget.com/2012/05/25/leap-motion-gesture-control-technology-hands-on>
- [3] Openlab.com.au. *Build ideas - Leap Motion* [online]. [cit. 2015-03-28]. Dostupné z: <http://openlab.com.au/shop/leap-motion/>
- [4] LEAP. *Leap Motion Developer portal* [online]. [cit. 2015-03-28]. Dostupné z: <https://developer.leapmotion.com/documentation/index.html>
- [5] Unity 3D. *Game engine* [online]. [cit. 2015-03-28]. Dostupné z: <http://unity3d.com/>
- [6] LEAP. *Leap Motion blog* [online]. [cit. 2015-03-28]. Dostupné z: <http://blog.leapmotion.com/rethinking-menu-design-in-the-natural-interface-wild-west/>
- [7] CDR.cz. *Leap Motion klepe na dveře*. [online]. 2013 [cit. 2015-03-28]. Dostupné z: <http://cdr.cz/clanek/leap-motion-klepe-na-dvere-jak-pohybove-ovladani-spolupracuje-s-windows-8-video>
- [8] monleap.fr. *Le kit de développement Leap Motion en photos* [online]. [cit. 2015-03-28]. Dostupné z: <http://monleap.fr/537-kit-developpement-leap-motion-photos/>
- [9] hanselman.com. *Leap Motion: Amazing, Revolutionary, Useless* [online]. 2013 [cit. 2015-03-28]. Dostupné z: <http://www.hanselman.com/blog/LeapMotionAmazingRevolutionaryUseless.aspx>
- [10] theprezenter.com. *Leap into Prezi* [online]. [cit. 2015-03-28]. Dostupné z: <http://theprezenter.com/hints-and-tips/prezi-and-leap-motion>

- [11] sparkfun.com.*Leap Motion Teardown* [online]. [cit. 2015-03-28]. Dostupné z: <https://learn.sparkfun.com/tutorials/leap-motion-teardown>
- [12] LEAP.*Leap Motion virtual reality* [online]. [cit. 2015-03-28]. Dostupné z: <https://www.leapmotion.com/product/vr>
- [13] LEAP.*Leap Motion virtual reality* [online]. [cit. 2015-03-28]. Dostupné z: <http://blog.leapmotion.com/osvr-hacker-dev-kit-will-feature-embedded-leap-motion-technology/>
- [14] LEAP.*Leap Motion developer portal* [online]. [cit. 2015-03-28]. Dostupné z: <https://developer.leapmotion.com/gallery/planetarium>
- [15] LEAP.*Leap App Store - Solar Walk* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/solar-walk-3d-solar-system-model/osx>
- [16] LEAP.*Leap App Store - CrystlViewer* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/crystalviewer/osx>
- [17] LEAP.*Leap App Store - Frog Dissection* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/frog-dissection/windows>
- [18] LEAP.*Leap App Store - 3D Geometry* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/3d-geometry/windows>
- [19] LEAP.*Leap App Store - Skywriting Alphabets* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/skywriting-alphabets/windows>
- [20] LEAP.*Leap App Store - Curious Kids* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/curious-kids/windows>
- [21] LEAP.*Leap App Store - Pointable* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/pointable/windows>
- [22] LEAP.*Leap App Store - Touchless For Windows* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/touchless-for-windows/windows>
- [23] LEAP.*Leap App Store - Touchless For Windows* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/airinput-trial/windows>

- [24] Unity3DStudentt.com. *Unity 3D Video Tutorials* [online]. [cit. 2015-03-28]. Dostupné z: <http://www.unity3dstudent.com/>
- [25] Unity3D.com. *Unity - Manual: Lighting Overview* [online]. [cit. 2015-03-28]. Dostupné z: <http://docs.unity3d.com/Manual/Lighting.html>
- [26] Unity3D.com. *Unity - Manual: Cookies* [online]. [cit. 2015-03-28]. Dostupné z: <http://docs.unity3d.com/Manual/Cookies.html>
- [27] Unity3D.com. *Unity - Manual: Unity Manual* [online]. [cit. 2015-03-28]. Dostupné z: <http://docs.unity3d.com/Manual/>
- [28] Unity3D.com. *Unity - Assets Store* [online]. [cit. 2015-03-28]. Dostupné z: <https://www.assetstore.unity3d.com>
- [29] Unity3D.com. *Unity - SpaceBuilder: Genesis* [online]. [cit. 2015-03-28]. Dostupné z: <https://www.assetstore.unity3d.com/en/#!/content/22175>
- [30] LEAP. *Leap App Store - Game Wave* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/gamewave/windows>
- [31] LEAP. *Leap App Store - Mudra Mouse* [online]. [cit. 2015-03-28]. Dostupné z: <https://apps.leapmotion.com/apps/mudra-mouse/windows>
- [32] imgkid.com. *Saturn Texture Map* [online]. [cit. 2015-03-28]. Dostupné z: <http://imgkid.com/saturn-texture-map.shtml>
- [33] galleryhip.com. *Uranus Planet Texture* [online]. [cit. 2015-03-28]. Dostupné z: <http://galleryhip.com/uranus-planet-texture.html>
- [34] planetpixemporium.com. *JHT's Planetary Pixel Emporium* [online]. [cit. 2015-03-28]. Dostupné z: <http://planetpixemporium.com/index.php>
- [35] Unity3D.com. *Unity - Scripting API* [online]. [cit. 2015-03-28]. Dostupné z: <http://docs.unity3d.com/Manual/index.html>
- [36] Unity3D.com. *Unity - Scripting API: MonoBehaviour* [online]. [cit. 2015-03-28]. Dostupné z: <http://docs.unity3d.com/ScriptReference/MonoBehaviour.html>

- [37] Answers Unity3D.com. *Answers Unity - How to raycast against GUI objects* [online]. [cit. 2015-03-28]. Dostupné z: <http://answers.unity3d.com/questions/821590/unity-46-how-to-raycast-against-ugui-objects-from.html>
- [38] LEAP. *Leap Legacy API Examples* [online]. [cit. 2015-03-28]. Dostupné z: <https://developer.leapmotion.com/forums/forums/7/topics/leap-legacy-api-examples>
- [39] Youtube.com. *Unity - Live Training 13th October 2014 - UI Panels, Panes & Windows* [online]. [cit. 2015-03-28]. Dostupné z: <https://www.youtube.com/watch?v=0L-Y8dPJHDM>
- [40] Unity3D.com. *Unity - Assets Store: Unity Samples: UI* [online]. [cit. 2015-03-28]. Dostupné z: <https://www.assetstore.unity3d.com/en/#!/content/25468>
- [41] iconarchive.com. *Iconset: Solar System Icons by Dan Wiersema* [online]. [cit. 2015-03-28]. Dostupné z: <http://www.iconarchive.com/show/solar-system-icons-by-dan-wiersma.html>
- [42] ndiscovered.com/. *NDISCOVERED - Multimedia design* [online]. [cit. 2015-03-28]. Dostupné z: <http://ndiscovered.com/>
- [43] tracertong.co.uk. *Coalition Font — Tracer Tong Font Works* [online]. [cit. 2015-03-28]. Dostupné z: <http://tracertong.co.uk>
- [44] Aldebaran.cz. *Aldebara home* [online]. [cit. 2015-03-28]. Dostupné z: <http://aldebaran.cz/>
- [45] Solar System Exploration. *Nasa - Solar System Exploration* [online]. [cit. 2015-03-28]. Dostupné z: <https://solarsystem.nasa.gov/index.cfm>
- [46] Unity3D. *Epic Orchestral Free Music Pack by Richard Daskas* [online]. [cit. 2015-03-28]. Dostupné z: <https://www.assetstore.unity3d.com/en/#!/content/34417>
- [47] Unity3D. *Euphorium Suite by Stephen Lu* [online]. [cit. 2015-03-28]. Dostupné z: <https://www.assetstore.unity3d.com/en/#!/content/22673>

[48] Unity3D. *Stars by Stephen Lu* [online]. [cit. 2015-03-28]. Dostupné z:
<https://www.assetstore.unity3d.com/en/#!/content/17497>

Seznam obrázků

1	Leap Motion [3]	13
2	Leap Motion - historický vývoj [7]	14
3	Leap Motion - rozebraný [11]	15
4	Leap Motion - podporované jazyky a vývojová prostředí [4]	15
5	Leap Motion - poloměr koule kterou ruka opisuje [4]	17
6	Leap Motion - směr os [2]	18
7	Leap Motion - gesta: kroužení, stiknutí klávesy, ťuknutí, švih [4]	18
8	Leap Motion - gesta: ťuknutí [9]	21
9	Leap Motion - budoucnost? [9]	21
10	Leap Motion - Leap Motion a Oculus Rift [9]	22
11	Leap Motion - OSVR [13]	23
12	Leap Motion - Oculus Rift a planetárium[14]	23
13	Aplikace Sluneční soustava pro Leap Motion	24
14	Leap Motion - Ukázka ovládání	25
15	Cyber Science - Motion	27
16	Form & Function 3D a Cyber Science - Motion: Zoology	28
17	Aplikace Solar Walk[15]	28
18	Aplikace CrystalViewer[16]	29
19	Frog Dissection[17]	29
20	3D Geometry[18]	30
21	Skywriting Alphabets a Curious Kids[19][20]	30
22	Leap Motion - rozšíření pro Scratch editor	31
23	Touchless for Windows[22]	32
24	Touchless for Windows[22]	33
25	Touchless for Windows[22]	33
26	Airinput Trial[23]	34
27	Airinput Trial	35
28	Pointable[21]	35
29	Pointable - nastavení	36
30	Pointable - pozicování oken	37
31	Pointable - nabídka programu	37

32	Mudra mouse[31]	38
33	Mudra mouse - nastavení[31]	38
34	Oslovila vás představená technologie Leap Motion?	40
35	Z celkového počtu 20 dotázaných odpovědělo 19, kladně pak 11 respondentů.	41
36	Líbila se vám nabídka aplikací?	41
37	Přišlo vám ovládání vyzkoušených aplikací intuitivní?	42
38	Použili byste zařízení Leap Motion ve své výuce?	42
39	Okno Unity editoru	47
40	Okno Unity editoru - tvorba základních 3D objektů	47
41	Směrové světlo - Cucoloris	48
42	Vliv veřejných proměnných na zobrazení v Unity	50
43	Animátor, animace, klíčování	50
44	Pozice pivotu: na levo (0.5 , 0.5) v pravo (0, 0)	52
45	Vlastnosti tlačítka	53
46	Pinch gesto	56
47	Slunce s panelem a zeleným indikátorem kliknutí	57
48	Pohled na Slunce z Venuše	59
49	Obrazovka intra	60
50	Pohled na sluneční soustavu se zvětšenými planetami	61
51	Porovnávací scéna	62
52	Porovnávací scéna	63
53	Dialogové okno pro sestavení aplikace	64
54	Změna UI pro ovládání klávesnicí a myší	65
55	Myšlenková mapa	74

8 Přílohy



Obrázek 55: Myšlenková mapa

8.1 Vybrané zdrojové kódy

8.1.1 Vykreslení kurzoru dle pozice ukazováčku

Kód pro vykreslení kurzoru v závislosti na souřadnicích prstu [38].

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;
using Leap;
using UnityEngine.EventSystems;
using System.Collections;
using System.Collections.Generic;

//https://developer.leapmotion.com/forums/forums/7/topics/leap-legacy-api-examples
// Draws the fingers onto the screen.
public class LeapFingerDraw : MonoBehaviour
{
    Controller m_leapController;
    public Texture2D m_noTouch;
    public Texture2D m_touch;
    public Texture2D m_borderTexture;

    public Camera UICam;
    public Camera MainCam;

    string buttonName;
    string objectName;

    bool wasClicked = false;
    bool wasClickedOnPlanet = false;

    void Start ()
    {
        m_leapController = new Controller ();
    }
}
```

```
}
```

```
    // visual feedback for finger positions
void OnGUI ()
{
    Frame frame = m_leapController.Frame ();
    InteractionBox interactionBox = frame.InteractionBox;
    Finger finger = frame.Hands.Frontmost.Fingers.Frontmost;

    if (frame.Pointables.Extended ().Count == 1) {
        if (frame.Pointables [0].Hand.Fingers.Extended () [0].Type () ==
            Finger.FingerType.TYPE_INDEX) {
            for (int i = 0; i < frame.Pointables.Extended().Count; ++i) {
                //Debug.Log (frame.Pointables
                    [i].Hand.Fingers.Extended() [0].Type());
                Vector3 screenPos = interactionBox.NormalizePoint
                    (frame.Pointables [i].StabilizedTipPosition).ToUnity ();

                float x = screenPos.x * UnityEngine.Screen.width;
                // y is flipped in GUI space.
                float y = (1 - screenPos.y) * UnityEngine.Screen.height;

                // compute the size scalar of the finger circle.
                float touchScalar = Mathf.Clamp (frame.Pointables
                    [i].TouchDistance, -0.05f,
                        frame.Pointables [i].TouchDistance);
                float size = 100.0f * Mathf.Sqrt (Mathf.Min (Mathf.Max
                    (Mathf.Abs (touchScalar), 0.001f), 0.9f));

                Vector2 normXY = new Vector2 (frame.Pointables [i].TipVelocity.x,
                    frame.Pointables [i].TipVelocity.y);

                // non uniform scaling for ellipsoid effect when finger is
```

```

        moving.
float yScale = 1.0f + Mathf.Min (0.6f, normXY.magnitude /
    2000.0f);
float xScale = Mathf.Sqrt (1.0f / yScale);
Matrix4x4 mat = GUI.matrix;
float angle = Mathf.Rad2Deg * Mathf.Atan (normXY.x / normXY.y);
// rotate the ellipsoid based on direction of the finger
    velocity.
GUIUtility.RotateAroundPivot (angle, new Vector2 (x, y));
if (frame.Pointables [i].TouchDistance > 0) {
    if(wasClicked){
        Button button = GameObject.Find
            (buttonName).GetComponent<Button> ();
        button.onClick.Invoke ();
    }
    wasClicked = false;
    if(wasClickedOnPlanet){
        ObjectClicked oc = GameObject.Find
            (objectName).GetComponent<ObjectClicked> ();
        oc.SetActive();
    }
    wasClickedOnPlanet = false;

float outerSizeX = ((size + 10) * xScale);
float outerSizeY = ((size + 10) * yScale);
GUI.DrawTexture (new Rect (x - outerSizeX / 2, y - outerSizeY
    / 2, outerSizeX, outerSizeY),
                m_borderTexture, ScaleMode.StretchToFill,
                true);
float innerSizeX = (size * xScale);
float innerSizeY = (size * yScale);
GUI.DrawTexture (new Rect (x - innerSizeX / 2, y - innerSizeY
    / 2, innerSizeX, innerSizeY),
                m_noTouch, ScaleMode.StretchToFill, true);
} else {

```

```

float scaleX = (size * xScale + 10);
float scaleY = (size * yScale + 10);
GUI.DrawTexture (new Rect (x - scaleX / 2, y - scaleY / 2,
    scaleX, scaleY),
    m_touch, ScaleMode.StretchToFill, true);

//bylo kliknuto?
RaycastHit hit;
Ray rayTest = MainCam.ViewportPointToRay(screenPos);
if(Physics.Raycast(rayTest, out hit)){
    objectName = hit.transform.gameObject.name;
    wasClickedOnPlanet = true;
}
PointerEventData pointer = new
    PointerEventData(EventSystem.current);
pointer.position = UICam.ViewportToScreenPoint(screenPos);
List<RaycastHit> rayResult = new List<RaycastHit>();
EventSystem.current.RaycastAll(pointer, rayResult);

if(rayResult.Count >0){
    Debug.Log(Time.deltaTime);
    buttonName =
        rayResult[0].gameObject.transform.parent.gameObject.name;
    wasClicked = true;
}
}
GUI.matrix = mat;
}
}
}
}
}

```

8.1.2 Pohyb ve scéně pomocí dlaně

```
using System;
using Leap;
using UnityEngine;
using UnityEngine.UI;
using System.Collections;
using MatrixMath;
using GlobalParameter;

public class MovingCameraByLeap: MonoBehaviour
{

    float speed = 0.2f;
    float speedUD = 0.1f;
    Controller controller;
    bool m_handOpenThisFrame = false;
    //bool m_handOpenLastFrame = false;
    bool m_handCloseThisFrame = false;

    //http://callmeavarice.blogspot.cz/2013/08/creating-leapmotion-gestures.html

    private bool fingersPinched = false;
    //private bool fingersPinchedDraging = false;
    private float pinchStartZPosition = 0f;
    private float pinchEndZPosition = 0f;

    private float FINGERS_PINCHED_X_AXIS_THRESHOLD = 20.5f;
    private float FINGERS_PINCH_RELEASED_X_AXIS_THRESHOLD = 6.5f;
    private int PREVIOUS_FRAME_ID = 1;

    //private bool palmFBMovement = false;
```



```

private bool palmDragging = false;
private float palmStartZPosition = 0f;
private float palmEndZPosition = 0f;

void Start(){
    controller = new Controller ();
}

// gets the hand furthest away from the user (closest to the screen).
Hand GetForeMostHand() {
    Frame f = controller.Frame();
    Hand foremostHand = null;
    float zMax = -float.MaxValue;
    for(int i = 0; i < f.Hands.Count; ++i) {
        float palmZ = f.Hands[i].PalmPosition.ToUnityScaled().z;
        if (palmZ > zMax) {
            zMax = palmZ;
            foremostHand = f.Hands[i];
        }
    }

    return foremostHand;
}

void rotateRL(Hand hand) {
    float rotThreshold = 0.0f;
    float handX = hand.PalmPosition.ToUnityScaled ().x;

    if (Mathf.Abs (handX) > rotThreshold) {
        transform.Rotate (Vector3.up, handX * 2.0f);
    }
}

```

```

}

void rotateUpDown(Hand hand) {

    float HandNormal = hand.Direction.Pitch;

    if (HandNormal > 0.6f) {
        transform.Rotate (Vector3.left, HandNormal * 2.0f);
    }
    if (HandNormal < 0) {
        transform.Rotate (Vector3.left, HandNormal * 2.0f);
    }

}

void rotateY(Hand hand) {

    float HandNormal = hand.PalmNormal.Roll;

    if (HandNormal < -0.6f) {
        transform.Rotate (Vector3.back, HandNormal * 2.0f);
    }
    if (HandNormal > 0.4f) {
        transform.Rotate (Vector3.back, HandNormal * 2.0f);
    }

}

void moveUpDown(Hand hand) {
    float handY = hand.PalmPosition.ToUnityScaled ().y;

    if (handY > 4.5f || handY < 0.8f) {

```

```

        speedUD = 30f;
    } else {
        speedUD = 0.1f;
    }

    if (handY > 3.5f ) {
        transform.position += transform.up * speedUD;
    }

    if (handY < 1.3f) {
        transform.position -= transform.up * speedUD;
    }
}

void moveFB(Hand hand)
{
    float handZ = hand.PalmPosition.ToUnityScaled ().z;

    if (Mathf.Abs(handZ) > 0.2f)
        speed = 0.2f;
    if (Mathf.Abs(handZ) > 0.8f)
        speed = 1.0f;
    if (Mathf.Abs (handZ) > 1.0f)
        speed = 3f;
    if (Mathf.Abs (handZ) > 1.2f)
        speed = 5f;
    if (Mathf.Abs (handZ) > 1.7f)
        speed = 50f;
    Debug.Log (speed.ToString ());
    if (handZ > 0.2f) {
        transform.position += transform.forward *speed;
    }
    if (handZ < -1.0f) {

```

```

        transform.position -= transform.forward *speed;

    }

}

void moveFBbyPinch(Hand hand)
{
    Finger thumb = hand.Fingers.Leftmost;
    Finger index = hand.Fingers.Frontmost;
    float pos1 = thumb.TipPosition.x;
    float pos2 = index.TipPosition.x;

    Vector thumbPosition = thumb.TipPosition;
    Vector indexPosition = index.TipPosition;

    float pinchDist = pos2 - pos1;

    if (pinchDist <= FINGERS_PINCHED_X_AXIS_THRESHOLD) {
        if (!fingersPinched) {
            fingersPinched = true;
            pinchStartZPosition = indexPosition.z;
            pinchEndZPosition = pinchStartZPosition;
        } else {
            int indexFingerId = index.Id;
            float currentZPosition = indexPosition.z;
            Frame previousFrame = controller.Frame (PREVIOUS_FRAME_ID);

            if (previousFrame != null && previousFrame.IsValid) {
                Finger indexFingerLastFrame = previousFrame.Finger
                    (indexFingerId);

                if (indexFingerLastFrame != null &&
                    indexFingerLastFrame.IsValid) {

```

```

        float indexfingerLastFrameZPosition =
            indexFingerLastFrame.TipPosition.z;
        float indexFingerPositionDelta = currentZPosition -
            indexfingerLastFrameZPosition;
        //moved away from the screen
        if (indexFingerPositionDelta > 0) {
            pinchEndZPosition = currentZPosition;
            //moving towards screen
        } else {
            if (currentZPosition < pinchStartZPosition) {
                pinchEndZPosition = currentZPosition;
            } else {
                pinchEndZPosition = pinchStartZPosition;
            }
        }
    }
}
}
} else if (fingersPinched && pinchDist >
    (FINGERS_PINCHED_X_AXIS_THRESHOLD +
    FINGERS_PINCH_RELEASED_X_AXIS_THRESHOLD)) {
    Debug.Log ("FINGERS PINCH RELEASED");
    fingersPinched = false;
    pinchEndZPosition = indexPosition.z;
    Debug.Log ("Distnce pinched: " + (pinchEndZPosition -
        pinchStartZPosition));
    transform.position += transform.forward * (pinchEndZPosition -
        pinchStartZPosition) * 0.5f;
}
}

void moveFBByClosedHand(Hand hand){
    Vector handPosition = hand.PalmPosition;
    if (IsHandFullClosed(hand)) {

```

```

if (!palmDragging) {
    palmDragging = true;
    Debug.Log ("Palm dragging");
    palmStartZPosition = handPosition.z;
    palmEndZPosition = palmStartZPosition;
} else {
    float currentZPosition = handPosition.z;
    Frame previousFrame = controller.Frame (PREVIOUS_FRAME_ID);
    if (previousFrame != null && previousFrame.IsValid) {
        Hand handLastFrame = previousFrame.Hands.Frontmost;
        if (handLastFrame != null && handLastFrame.IsValid) {
            float handLastFrameZPosition = handLastFrame.PalmPosition.z;
            float handPalmPositionDelta = currentZPosition -
                handLastFrameZPosition;

            if (handPalmPositionDelta > 0) {
                palmEndZPosition = currentZPosition;
            } else {
                if (currentZPosition < palmStartZPosition) {
                    palmEndZPosition = currentZPosition;
                } else {
                    palmEndZPosition = palmStartZPosition;
                }
            }
        }
    }
}

else if(palmDragging && !IsHandFullClosed(hand)){
    Debug.Log("End palm dragging");
    palmDragging = false;
    palmEndZPosition = handPosition.z;

    Debug.Log ("Distance pinched: " + (palmEndZPosition -
        palmStartZPosition));
}

```

```

        transform.position += transform.forward * (palmEndZPosition -
            palmStartZPosition)*2f;
    }
}

bool IsHandOpen(Hand h) {
    return h.Fingers.Extended ().Count == 5;
}

bool IsHandFullClosed(Hand h) {
    return h.Fingers.Extended ().Count == 0;
}

void Update () {
    Hand foremostHand = controller.Frame ().Hands.Frontmost;
    m_handOpenThisFrame = IsHandOpen(foremostHand);
    m_handCloseThisFrame = IsHandFullClosed (foremostHand);
    if (foremostHand != null && m_handOpenThisFrame) {
        rotateRL (foremostHand);
        rotateUpDown (foremostHand);
        moveUpDown (foremostHand);
        moveFB(foremostHand);
        rotateY (foremostHand);

    } else if (foremostHand != null ) {
        moveFBbyPinch(foremostHand);
        //moveFBByClosedHand(foremostHand);
    }
}
}
}

```

8.1.3 Gesta

```
using System;
using Leap;
using UnityEngine;
using UnityEngine.UI;
using System.Collections;
using MatrixMath;
using GlobalParameter;

public class GestureHands: MonoBehaviour
{
    Controller controller;

    void Start(){
        controller = new Controller ();
        controller.EnableGesture (Gesture.GestureType.TYPECIRCLE);
        controller.EnableGesture (Gesture.GestureType.TYPESWIPE);

        controller.Config.SetFloat("Gesture.Circle.MinRadius", 20.0f);
        controller.Config.SetFloat("Gesture.Circle.MinArc", 4f);
        controller.Config.SetFloat("Gesture.Swipe.MinLength", 150.0f);
        controller.Config.SetFloat("Gesture.Swipe.MinVelocity", 500f);
        controller.Config.Save();
    }

    void Update(){
        Frame frame = controller.Frame ();
        GestureList gl = frame.Gestures ();

        Hand foremostHand = null;
        float zMax = -float.MaxValue;
        for(int i = 0; i < frame.Hands.Count; ++i) {
            float palmZ = frame.Hands[i].PalmPosition.ToUnityScaled().z;
            if (palmZ > zMax) {
```



```

        zMax = palmZ;
        foremostHand = frame.Hands[i];
    }
}
FingerList fl = foremostHand.Fingers;
Finger finger = foremostHand.Fingers.Frontmost;

for (int i = 0; i < gl.Count; i++) {
    Gesture gesture = gl[i];

    //planet scaling by ginger rotation
    if(gesture.Type == Gesture.GestureType.TYPE_CIRCLE && finger.Type()
        == Finger.FingerType.TYPE_INDEX && finger.IsExtended &&
        fl.Extended().Count == 1)
    {
        CircleGesture circleG = new CircleGesture(gesture);

        Vector circleDirection = frame.Pointables.Frontmost.Direction;
        float angle = circleDirection.AngleTo(circleG.Normal);
        Debug.Log(SpaceObjectsScaling.planetScale);
        if(SpaceObjectsScaling.planetScale2 <= 50.0){

            if(angle<= (3.14/4)&& SpaceObjectsScaling.planetScale2 != 50.0){
                SpaceObjectsScaling.planetScale2 += 1f;
            }else if(angle >= (3.14/4)) {
                double isSmall = SpaceObjectsScaling.planetScale-1.0;
                if(isSmall >= 2.0)
                    SpaceObjectsScaling.planetScale2 -= 1f;
            }
        }
    }
    if(gesture.Type == Gesture.GestureType.TYPE_CIRCLE &&
        fl.Extended().Count == 2)
    {
        CircleGesture circleG = new CircleGesture(gesture);
    }
}

```

```
Vector circleDirection = frame.Pointables.Frontmost.Direction;
float angle = circleDirection.AngleTo(circleG.Normal);
Debug.Log(SpaceObjectsScaling.timeSpeedUp2);
if(SpaceObjectsScaling.timeSpeedUp2 <= 451365){

    if(angle<= (3.14/4)&& SpaceObjectsScaling.timeSpeedUp2 < 450000){
        SpaceObjectsScaling.timeSpeedUp2 += 1000;
    }else if(angle >= (3.14/4)) {
        double isSmall = SpaceObjectsScaling.timeSpeedUp2-1000;
        if(isSmall >= 2.0)
            SpaceObjectsScaling.timeSpeedUp2 -= 1000;
    }
}
}
if(gesture.Type == Gesture.GestureType.TYPE_SWIPE)
{
    SwipeGesture swipe = new SwipeGesture(gesture);
    Vector swipeDirection = swipe.Direction;

    if(swipeDirection.x < 0){
        Debug.Log("Left");
    }
    else if(swipeDirection.x > 0)
    {
        Debug.Log("Right");
    }
}
}
}
}
}
}
```

8.2 Dotazník

Využití bezdotykové technologie Leap Motion při ovládání interaktivních aplikací ve výuce.

Dobrý den, v rámci své diplomové práce provádím výzkum využití a podpory výuky pomocí bezdotykového ovladače Leap Motion. Zjišťuji, zda a jakým způsobem by mohla být tato technologie využívána ve výuce jako interaktivní doplněk, či jako kompenzační pomůcka. Budu velmi vděčný za Vaši pomoc a vyplnění tohoto **anonymního** dotazníku.

Pokyny k vyplnění:

Dotazník se skládá z 12 otázek. Každá otázka, u které jsou uvedené možnosti má právě jednu možnou odpověď, pokud není u otázky určeno jinak. U otázek bez daných možností odpovězte svými slovy. Pokud se Vám otázka bude zdát nesrozumitelná, nejednoznačná nebo byste odpověděli jinak, můžete pod ni napsat libovolně dlouhou odpověď.

1. Jaké je vaše pohlaví?

- Muž
- Žena

2. Jaký je Váš věk?

- do 30 let
- 31 až 40 let
- 41 až 50 let
- více než 50 let

3. Jaká je délka praxe na pozici učitele?

- do 5 let
- 6 až 10 let
- 11 až 20 let
- více než 20 let

4. Jaký je Váš obor, popřípadě kombinace oborů?

5. Na jaké škole učíte?

- 1. stupeň ZŠ
- 2. stupeň ZŠ
- střední škola
- vysoká škola

6. Používáte interaktivní prvky ve výuce? (Možno označit více odpovědí)

- interaktivní tabule
- tablety
- hlasovací zařízení
- audiovizuální techniku
- jiné (napište jaké)

7. Jak často využíváte interaktivní zařízení?

- každou hodinu
- téměř na každé hodině
- příležitostně
- velmi výjimečně

8. Oslovila vás představená technologie Leap Motion?

Ano

Ne

9. Viděli byste potenciál ve využití Leap Motion ve výuce?

Ano

Ne

10. Co byste u této technologie ocenili, nebo naopak zkritizovali?

11. Líbila se vám nabídka aplikací?

Ano

Ne

12. Přišlo vám ovládání vyzkoušených aplikací intuitivní?

Ano

Ne

13. Použili byste zařízení Leap Motion ve své výuce?

Ano

Ne

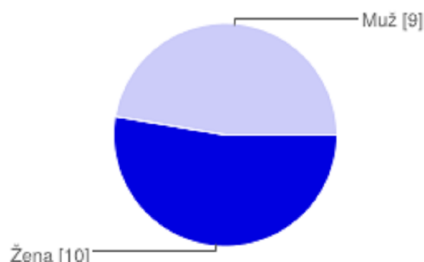
Elektronická verze dotazníku:

<http://goo.gl/forms/EFWshAkrJb>

Počet odpovědí: 20

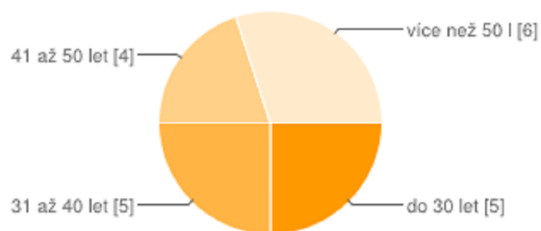
Souhrn

1. Jaké je vaše pohlaví?



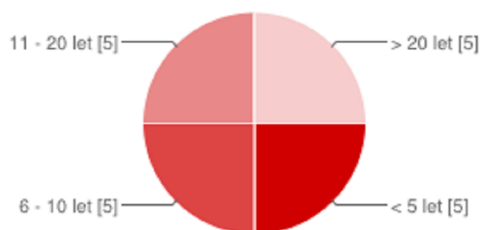
Žena	10	50 %
Muž	9	45 %

2. Jaký je váš věk?



do 30 let	5	25 %
31 až 40 let	5	25 %
41 až 50 let	4	20 %
více než 50 let	6	30 %

3. Jaká je délka vaší praxe?



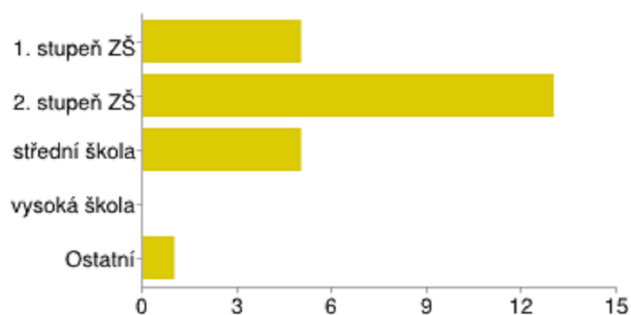
< 5 let	5	25 %
6 - 10 let	5	25 %
11 - 20 let	5	25 %
> 20 let	5	25 %

4. Jaký je váš obor, popřípadě kombinace oborů?

M - Zpv

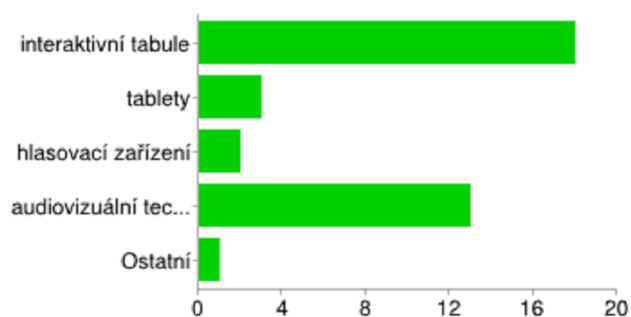
M-AJ-ICT
M - ICT
Př - TV
Př - Ch
M - VT
F - IT
Nj - Občanka
Tv - Inf

5. Na jaké škole učíte?



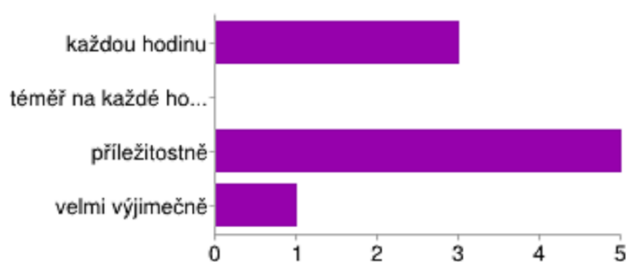
1. stupeň ZŠ	5	25 %
2. stupeň ZŠ	13	65 %
střední škola	5	25 %
vysoká škola	0	0 %
Ostatní	1	5 %

6. Používáte interaktivní prvky ve výuce? (Možno označit více odpovědí)



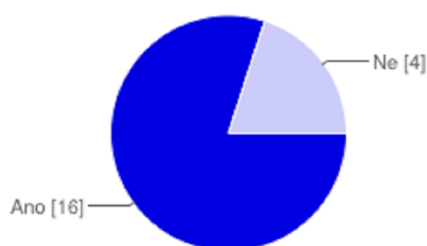
interaktivní tabule	18	90 %
tablety	3	15 %
hlasovací zařízení	2	10 %
audiovizuální technika	13	65 %
Ostatní	1	5 %

7. Jak často využíváte interaktivní zařízení?



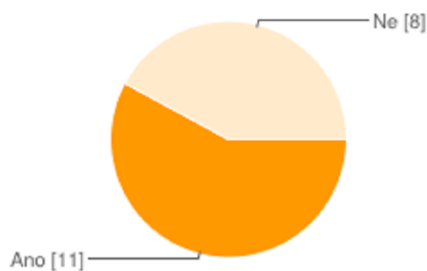
každou hodinu	3	15 %
téměř na každé hodině	0	0 %
příležitostně	5	25 %
velmi výjimečně	1	5 %

8. Oslovila vás představená technologie Leap Motion?



Ano	16	80 %
Ne	4	20 %

9. Viděli byste potenciál ve využití Leap Motion ve výuce?



Ano	11	55 %
Ne	8	40 %

10. Co byste u této technologie ocenili, nebo naopak zkritizovali?

+ přiblížení ovládání realitě - Náročnost na přesnost "manipulace"

- je to nové, takže si myslím, že by to žáky oslovilo líbila se mi aplikace Sluneční soustava, lebka

- těžší ovládání pro začátečníka - nedostatečné množství SW pro výuku různých předmětů
Lze použít jako doplněk výuky, ale vzhledem k ceně si myslím, že by se jednalo o velký komfort

novinka, která může zaujmout studenty "elegance" ovládání

Zatím si nedovedu představit, v čem bych toto zařízení využil (možná ve stereometrii)

jednoduchost ovládání (po zvládnutí) plynulost ovládání aplikací

Zajímavé pro rozšíření výuky

Nemáme k tomu schopné žáky

- softwarové zázemí - ovládání

- nedostatek výukových materiálů + jednoduchost použití

- Poměrně náročné ovládání, možnost testovat? + názornost aplikací (lebka ...), interaktivita

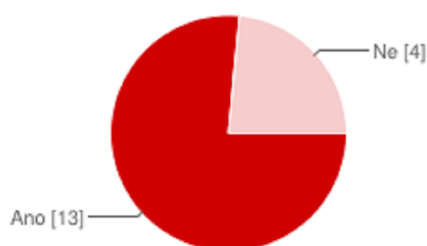
+ Postiženým žákům umožní lepší ovládání - mě, jako neohrabanou učitelku značně

znervózní, pokud se hned nepřesunu tam, kam mám

- velká ztráta času (naučení ovládání), kvůli chvilce zábavy + vhodný jako doplněk + určitě

má potenciál

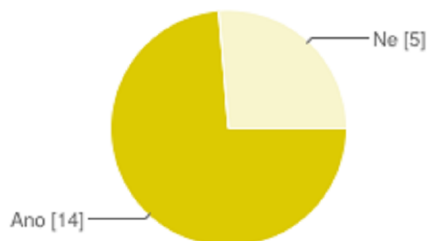
11. Líbila se vám nabídka aplikací?



Ano **13** 65 %

Ne **4** 20 %

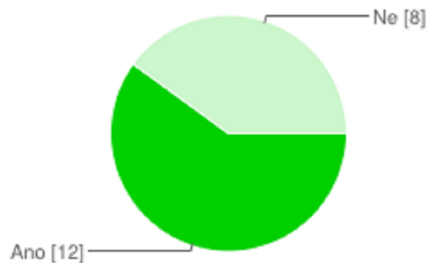
12. Přišlo vám ovládání vyzkoušených aplikací intuitivní?



Ano **14** 70 %

Ne **5** 25 %

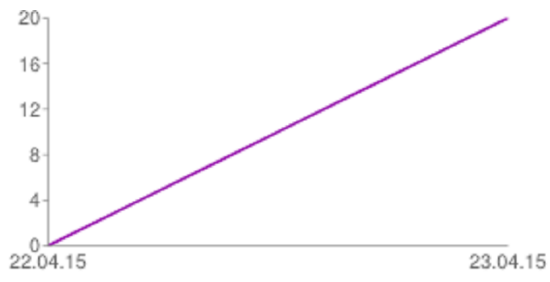
13. Použili byste zařízení Leap Motion ve své výuce?



Ano **12** 60 %

Ne **8** 40 %

Počet odpovědí za den



8.4 Příloha CD

Příloha na CD obsahuje text diplomové práce, vytvořenou aplikaci se zdrojovými soubory a dotazníkem.