

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Interaktivní virtuální prohlídka pro OS Android

Tomáš Kulík

© 2020 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Tomáš Kulík

Systemové inženýrství a informatika
Informatika

Název práce

Interaktivní virtuální prohlídka pro OS Android

Název anglicky

Interactive virtual tour for OS Android

Cíle práce

Cílem teoretické části bakalářské práce je analýza podkladů k implementaci mobilní aplikace pro virtuální prohlídku. Teoretická část se skládá z dílčích cílů. Prvním dílčím cílem je vysvětlení základních pojmů z oblasti virtuální reality a termínů s ní spojených (virtuální svět, imerze, interaktivita, zpětná vazba). Dalším dílčím cílem je hlubší analýza interaktivity, konkrétně jaký má význam a jak se vytváří. Závěrečným cílem je popis vývoje aplikace pro mobilní telefon s operačním systémem Android pomocí herního enginu Unity. Cílem praktické části bakalářské práce je samotná tvorba interaktivní virtuální prohlídky obchodu s hudebninami v herním enginu Unity pro operační systém Android.

Metodika

Metodika tvorby bakalářské práce je založena na studiu a analýze odborných informačních zdrojů z oblasti virtuální reality, interaktivity a herního enginu Unity. Pro vypracování práce bude prostudován herní engine Unity a limity vztahující se k vývoji virtuální prohlídky pro mobilní telefony. Následně bude popsán postup realizace řešení a dojde ke zhodnocení výsledků.

Doporučený rozsah práce

30 – 50 stran

Klíčová slova

vývoj mobilní aplikace, virtuální prohlídka, interaktivita, OS Android, Unity

Doporučené zdroje informací

KUKSA, Iryna a Mark CHILDS. Making sense of space: the design and experience of virtual spaces as a tool for communication. Oxford, UK: Chandos Publishing, an imprint of Elsevier Limited, 2014. ISBN 9781843347408

Unity – Manual: XR. [online]. © 2019 Unity Technologies. Publication [cit. 22.05.2019]. Dostupné z: <https://docs.unity3d.com/Manual/XR.html>

What is Virtual Reality (VR)? Ultimate Guide to Virtual Reality (VR) Technology. Reality Technologies (MR, AR, VR) News | Guides | Companies | Products [online]. © RealityTechnologies.com Diversified Internet Holdings LLC [cit. 22.05.2019]. Dostupné z: <https://www.realitytechnologies.com/virtual-reality/>

XR – Unity. [online]. © 2019 Unity Technologies [cit. 14.04.2019]. Dostupné z: <https://unity3d.com/learn/tutorials/s/xr>

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

Ing. Dana Vynikarová, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 2. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 2. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 03. 03. 2020

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Interaktivní virtuální prohlídka pro OS Android" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 23.03.2020

Poděkování

Rád bych touto cestou poděkoval paní Ing. Daně Vynikarové, Ph.D. za pomoc, trpělivost a věcné připomínky při vedení mé práce.

Interaktivní virtuální prohlídka pro OS Android

Abstrakt

Tato bakalářská práce se zabývá problematikou tvorby virtuální prohlídky v herním enginu Unity pro mobilní zařízení s operačním systémem Android. V teoretické části jsou popsány důležité termíny vztahující se k virtuální realitě jako je například, kyberprostor či imerze. Dále je popsána interaktivita a její dělení. Nakonec se krátce zmiňuje o použitých technologiích potřebných pro vytvoření virtuální prohlídky. V praktické části byla vytvořena virtuální prohlídka obchodu s hudebními nástroji pro mobilní platformu Android. Prohlídka využívá ovládací prvky, které uživateli umožňují s nástroji interagovat.

Klíčová slova: vývoj mobilní aplikace, virtuální prohlídka, interaktivita, OS Android, Unity

Interactive virtual tour for OS Android

Abstract

This bachelor thesis deals with problems of creation a virtual tour in the Unity game engine for a mobile device with Android operating system. Theoretical part describes important terms related to virtual reality such as cyberspace and immersion. After that, interactivity and its division is characterized. Finally, it briefly mentions the technologies used to create the virtual tour. In the practical part, a virtual tour of the musical instrument store for the Android mobile platform was created. The tour uses controls that allow the user to interact with the instruments.

Keywords: mobile application development, virtual tour, interactivity, OS Android, Unity

Obsah

1 Úvod	11
2 Cíl práce a metodika	12
2.1 Cíl práce.....	12
2.2 Metodika.....	12
3 Teoretická východiska	13
3.1 Stručná historie virtuální reality.....	13
3.2 Virtuální realita	14
3.3 Kyberprostor	14
3.4 Bezprostřednost.....	15
3.5 Imerze	15
3.6 Interaktivita.....	16
3.6.1 Manipulace	16
3.6.2 Navigace	18
3.6.3 Interakce s ostatními	19
3.6.4 Tok interakce	19
3.7 Technologie	20
3.7.1 Unity.....	21
3.7.2 Android.....	21
4 Vlastní práce	23
4.1 Příprava.....	23
4.2 Pohyb.....	24
4.3 Indikátor průběhu	26
4.4 Informace o nástroji	27
4.5 Příprava plátna pro zobrazení informací	28
4.6 Teleport k plátnu a vytvoření kopie nástroje	29
4.7 Třídění	31
5 Výsledky	35
6 Závěr	36
7 Seznam zdrojů	37
8 Přílohy.....	39
8.1 Instalační aplikační balíček a vytvořené skripty	39
8.2 Snímky z aplikace	39

Seznam obrázků

Obrázek 1 – Originální Oculus Rift [3].....	13
Obrázek 2 – Google Earth [9].....	20
Obrázek 3 – Nastavení pro spuštění aplikace na platformě Android.....	24
Obrázek 4 – Teleport skript jako komponenta.....	25
Obrázek 5 – Teleportační objekty v aplikaci.....	26
Obrázek 6 – Indikátor průběhu.....	26
Obrázek 7 – <i>InstrumentInfo</i> skript jako komponenta.....	28
Obrázek 8 – <i>FillCanvasWithInfo</i> skript jako komponenta.....	29
Obrázek 9 – Nevyplněné plátno.....	29
Obrázek 10 – Vyplněné plátno s informacemi a kopie hudebního nástroje.....	30
Obrázek 11 – <i>ObjectInteraction</i> skript jako komponenta.....	31
Obrázek 12 – Možnosti třídění.....	32
Obrázek 13 – Nástroje před vytríděním.....	34
Obrázek 14 – Nástroje po vytrídění.....	34
Obrázek 15 Rotující hudební nástroj.....	39
Obrázek 16 Plátno s informacemi o vybraném hudebním nástroji s rotující kopíí daného nástroje.....	40
Obrázek 17 Hudební nástroje z výšky.....	41
Obrázek 18 Hudební nástroje z pohledu uživatele.....	42

1 Úvod

Virtuální realita v dnešní době zažívá veliký rozmach. Využívá se v nespočetně odvětvích od armády a lékařství až po domácí využití jako jsou počítačové hry. Hardwarové technologie se posunuly do takové fáze, že může být virtuální svět stejně detailní jako ten reálný, a to i prostřednictvím mobilního telefonu. Existují i takzvané virtuální prohlídky, které umožňují uživatelům poznat nová zajímavá místa, například památky či jiná významná místa z pohodlí vlastního domova.

Důvod výběru tohoto tématu byl především zájem o virtuální realitu. Dále také zvědavost, zdali by vytvoření virtuální prohlídky pro místa, určené k prodeji zboží, mělo reálné využití v praxi. Na světě se již vyskytují virtuální prohlídky obchodních center, či obchodů, ale osobně se autor nesešel s prohlídkou s interaktivními prvky, která by umožnila v nějaké míře s prodáváním zbožím interagovat.

Tato práce se zabývá vytvořením virtuální prohlídky pro mobilní zařízení s operačním systémem Android. Do prohlídky se přidají ovládací prvky a umožní se tím interagovat s objekty. Pro prohlídku byl vymodelován obchod s hudebními nástroji.

V práci se čtenář seznámí s teoretickými východisky, která mu pomohou se sblížit se s tématem virtuální reality a následně i s interaktivitou. Dále se seznámí s technologiemi, které se budou pro vytvoření virtuální prohlídky používat. V praktické části pomocí nabitých poznatků se vytvoří aplikace pro mobilní systém Android.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem teoretické části bakalářské práce je analýza podkladů k implementaci mobilní aplikace pro virtuální prohlídku. Teoretická část se skládá z dílčích cílů. Prvním dílčím cílem je vysvětlení základních pojmů z oblasti virtuální reality a termínů s ní spojených (virtuální svět, imerze, interaktivita, zpětná vazba...). Dalším dílčím cílem je hlubší analýza interaktivity, konkrétně jaký má význam a jak se vytváří. Závěrečným cílem je popis vývoje aplikace pro mobilní telefon s operačním systémem Android pomocí herního engine Unity.

Cílem praktické části bakalářské práce je samotná tvorba interaktivní virtuální prohlídky obchodu s hudebninami v herním engine Unity pro operační systém Android.

2.2 Metodika

Metodika tvorby bakalářské práce je založena na studiu a analýze odborných informačních zdrojů z oblasti virtuální reality, interaktivity a herního engine Unity. Pro vypracování práce je prostudován herní engine Unity a limity vztahující se k vývoji virtuální prohlídky pro mobilní telefony. Následně je popsán postup realizace řešení a zhodnocení výsledků.

3 Teoretická východiska

3.1 Stručná historie virtuální reality

Virtuální realita není nový koncept. Začátky virtuální reality se datují již od roku 1835, kdy Sir Charles Wheatstone poprvé popsal termín stereoskopie, ten značí stav, kdy mozek přidá obrázku pocit hloubky. Nastane, pokud ke každému oku přiložíme obrázek stejného objektu vyobrazeného v rozdílných úhlech. Dalším velkým milníkem je příběh napsaný Stanleym Weinbaumem pod názvem *Pygmalion's Spectacles* [1] v roce 1935. Příběh je o hlavní postavě, jež se potká s profesorem, který vynalezl brýle, pomocí kterých se člověk dostane do filmu. Postava využívá všech svých smyslů a je protagonistou celého filmu. Toto dílo někteří lidé považují za počátek virtuální reality. K prvnímu velkému rozmachu došlo v roce 1985 založením VPL Research, Inc., první firmy, jenž se zabírá prodejem brýlí a rukavic pro virtuální realitu. Nadšení pro virtuální realitu na konci 80. let a začátku 90. let prudce rostlo, ale časem se narazilo na hardwarové limity tehdejší doby a popularita začala klesat. Po 20 letech zažila virtuální realita druhý rozmach, kdy Oculus Rift (viz Obrázek 1) získal na crowdfundingové platformě Kickstarter 2,4 milionů dolarů (cca 59,3 milionu Kč). O 2 roky později byla firma odkoupena Facebookem za 2 miliardy dolarů (cca 49,4 milionu Kč). Od této chvíle se čím dál více začalo investovat do virtuální reality [2].



Obrázek 1 – Originální Oculus Rift [3]

3.2 Virtuální realita

Termín **virtuální realita** byl poprvé zaveden německým filozofem Immanuelem Kantem, ale nebyl nijak spjat s technologií. Kant virtuální realitou popisoval realitu v mysli člověka na rozdíl od reálného světa. Moderní význam byl popularizován Jaronem Lanierem v 80. letech. Virtuální realitu lze definovat jako navození cíleného chování v organismu pomocí umělé smyslové stimulace, zatímco organismus má malé nebo nemá žádné povědomí o rušení z reálného světa. Termín **virtuální prostředí** často používají univerzitní vědci, ale můžeme to považovat za synonymum k virtuální realitě. Další termín, se kterým se můžeme často setkat je **rozšířená realita** (*augmented reality*), který se liší od té virtuální tím, že do reálného světa přidává počítačově vygenerované objekty, například přes mobilní telefon či tablet, a tím realitu rozšiřuje. Termín **smíšená realita** (*mixed reality*) se někdy používá k označení celého spektra, které zahrnuje virtuální realitu, rozšířenou realitu a běžnou realitu. Často se také používá pojem *telepresence* (systém, který uživatelům umožňuje cítit se, jako by byli jinde ve skutečném světě) a *teleoperation* (pokud jsou schopni něco kontrolovat, například létající dron). [4]

3.3 Kyberprostor

Kyberprostor je počítačem vytvořené navigovatelné nekonečno, které zdánlivě existuje za obrazovkou počítače a umožňuje propojit a odpojit jeho uživatele v reálném čase, když jsou aktivně zapojeni v síťové elektronické komunikaci. Kyberprostor zahrnuje velkou šíři 3D digitálních prostorů, jež mohou být nějakým způsobem spjaté s reálným světem nebo mohou být zcela imaginární. Například počítačové hry pro více hráčů nebo virtuální výuková centra. Tyto místa jsou interaktivní, navigovatelné v reálném čase a nejsou vázány fyzickými omezeními. Kyberprostor je považován za digitálně vytvořený paralelní kulturní svět, který vytváří velké množství příležitostí pro kreativitu, učení a komunikaci. V dnešní době existuje nepřehledný rozsah virtuálních světů, kde lidé mohou být kreativní, socializovat se či být součástí komunity počítačové hry pro více hráčů. Virtuální svět je možné definovat jako synchronní a stálou síť lidí a počítačových programů (vyobrazený avatary), která je řízená pomocí propojených počítačů a vytváří co nejuvěrohodnější prostor pro tyto uživatele. **Avatar** je grafické znázornění uživatele v prostředí, které je pod jeho přímou kontrolou. Avatarem obývané virtuální komunity byly vždy velmi populární pro komunikaci

a socializaci, ale používají se i například pro 3D rekonstrukci starobylých budov nebo historických etap na základě archeologických a vizuálních důkazů [5].

3.4 Bezprostřednost

Bezprostřednost (*immediacy*) je často označována jako absence vnímání technologie, která zprostředkovává danou činnost. Ve většině případů je hlavním cílem komunikačních technologií dosažení úplné bezprostřednosti. Znatelná přítomnost technologie je často považována za překážku k úplnému pocitu spojení s virtuálním prostorem nebo s ostatními uživateli uvnitř. Na druhé straně máme *hypermediacy*, kde je technologie vnímána a kde se zkoumá náš vztah k ní [5].

3.5 Imerze

Imerze (*immersion*) neboli také **ponoření** či **pohroužení**. Tento termín se může dále rozdělit na psychologickou imerzi (*psychological immersion*) a vjemovou imerzi (*perceptual immersion*).

Psychologická imerze popisuje pocit ponoření se do prostředí nebo zážitku. Například, když se díváme na televizní seriál a emocionálně se vcítíme do postav a jejich životů, nebo když hrajeme hru a ztratíme ponětí o čase do takové míry, že přestáváme vnímat své okolí. Psychologická imerze nemusí být nutně zaměnitelná s bezprostředností. To je možné uvést na příkladu virtuálního světa Second Life, hra má viditelné rozhraní, vyskakovací okna nebo se může dokonce i zasekávat. Avšak hráč si na tyto rušení do takové míry zvykl, že i přes to se do hry dokáže ponořit a tyto vady ignorovat.

Vjemová imerze na druhou stranu popisuje, do jaké míry je naše smyslové vnímání ovlivněné použitými technologiemi. Jako příklad kompletně percepčně imerzního prostředí můžeme uvést Cybersphere z Univerzity z Werwicku. Tato dutá konstrukce o průměru 3,5 m umožňuje pohyb návštěvníků uvnitř, a přitom snímá jejich pohyby a podle nich upravuje vnitřní projekce. Cybersphere dosahuje úplného vjemového ponoření a to tak, že zcela obklopuje uživatele a neobsahuje žádné rozhraní. I když vjemová imerze přispívá k psychologické, není jí vždy zapotřebí aby k psychologické imerzi došlo [5].

3.6 Interaktivita

Pojem **interaktivita** znamená v každém odvětví něco odlišného. Základním modelem, ze kterého vychází koncept sociologické interakce, je vztah mezi dvěma nebo více lidmi, kteří si v dané situaci vzájemně přizpůsobují své chování. Pojem interakce v mediálních a komunikačních studiích se často používá k označení akcí posluchačů nebo příjemců v souvislosti s mediálním obsahem. A v informatickém odvětví to označuje vztah mezi lidmi a stroji (nikoli však komunikaci mezi lidmi zprostředkovanými stroji) [6].

Interaktivitu je také možné definovat jako vzájemné působení nebo ovlivňování činitelů mezi sebou. K interakci s prostředím generovaným počítačem tedy dochází, když na uživatelské vstupy reaguje odpovídající akce počítače. Interakce s virtuálním světem je klíčovou součástí zážitku ve virtuální realitě. Pokud zobrazení virtuálního světa nereaguje alespoň na fyzický pohyb uživatele, nepovažujeme ji za virtuální realitu. Interakce s počítačem je závislá na tom, jak je navržené uživatelské rozhraní (*user interface*). **Uživatelské rozhraní** je mechanismus, kterým účastník interaguje s médiem. Lidé nejsou navrženi k interakci se stroji, a proto většina uživatelských rozhraní není pro neškoleného uživatele přirozená. Naštěstí se lidé dokážou naučit novým způsobům interakce se svým prostředím. Navíc se mohou některá rozhraní integrovat do kultury tak, že se stanou přirozenými, například rozhraní v automobilu – volant, pedály na kontrolu rychlosti a spojka. Uživatel při koupě nového auta nemusí podstoupit nový trénink, aby věděl, jak auto ovládat. [7]

Existují tři klíčové způsoby, jak uživatelé interagují s virtuálním světem prostřednictvím virtuální reality. Jsou to **manipulace, navigace a komunikace**.

3.6.1 Manipulace

Velká výhoda bytí v interaktivním virtuálním prostoru je schopnost interagovat či manipulovat objekty v tomto prostoru. Schopnost experimentovat v novém prostředí, reálném nebo virtuálním, pomáhá člověku zjistit, jak tento svět funguje. Manipulace umožňuje uživateli modifikovat svět a objekty, které v něm existují. Ve virtuálním světě manipulace probíhá ve dvou fázích. První je vybrán objekt a poté je na daný objekt vykonaná akce. Někdy se tyto fáze mohou provést najednou. Na rozdíl od reálného světa, ve virtuální realitě existuje celá řada metod pro manipulaci s virtuálními objekty [7].

3.6.1.1 Přímé ovládání uživatelem

Přímé ovládání uživatelem (*direct user control*) je metoda, ve které uživatel interaguje s objekty ve virtuálním světě způsobem v podstatě stejným jako v reálném světě. Většina interakcí této metody používá gesta či pohled k výběru. Jako příklad gesta je sevření ruky uživatele v pěst. To se interpretuje jako hmátnutí a objekt se přemístí do ruky a tu poté následuje, dokud gesto sevření není přerušeno. Zaměření se na objekt a poté mrknutí či zmáčknutí tlačítka je příkladem metody výběru pohledem. Je preferováno používání přímého ovládání, ale mohou se naskytnout případy, kdy uživatel upřednostňuje, aby se technologie ujímala složitějších úkonů, jako je například automatické seřazování objektů [7].

3.6.1.2 Fyzické ovládání

Fyzické ovládání (*physical control*) je takové, kdy se virtuální svět ovládá prostřednictvím zařízení z reálného světa. Uživatel získává hmatovou zpětnou vazbu mačkáním tlačítek a dalších činností. Mezi běžné fyzické ovládací prvky spadají tlačítka, prepínače, jezdcí (*slider*), joystick a tak dále. Při návrhu rozhraní, které využívá fyzické ovládání je nutné brát v potaz jeho vztah k virtuálnímu objektu. Design bude lépe fungovat, pokud fyzické ovládání bude úzce korespondovat s tím virtuálním. Například při manipulaci jednotky během tréninkové mise astronautů by bylo dobré mít sedačku a tlačítka na pozicích, které co nejdříve napodobují rozložení v reálném objektu [7].

3.6.1.3 Virtuální ovládání

Virtuální ovládání (*virtual control*) je ovládání, které se vyskytuje pouze ve virtuálním světě. Většina ovládacích prvků je počítačově vygenerovaná a vypadá velice podobně jako jsou fyzická tlačítka či prepínače. Existují důvody, proč dát přednost virtuálnímu ovládání před fyzickým. Například, když chceme zmenšit počet fyzických přístrojů. Ovládání může být schováno do doby, kdy jej potřebujeme a aktivujeme ho přes kliknutí, nebo např. změním vzhled ovládacích prvků abychom ukázali, zdali jsou aktivní (př. ovládání při neaktivitě zšedne). S virtuálním ovládáním potřebujeme nějakým způsobem interagovat. K tomuto účelu často slouží fyzický ovladač, který jej aktivuje. To má výhodu v tom, že fyzický ovladač potřebuje méně vstupních signálů a tím může využívat větší rozsah funkcí v rozhraní s virtuálním ovládáním [7].

3.6.1.4 Ovládání prostředníkem

Ovládání prostředníkem (*agent control*) využívá prostředníka, který provádí požadované pokyny, jež mu byly sděleny přes komunikaci s uživatelem. Prostředník může být jak člověk, tak počítačem řízená bytost. Komunikace s agentem může být hlasem nebo gesty [7].

3.6.2 Navigace

Navigace popisuje pohyb z místa na místo. Ve skutečném světě k navigaci využíváme chůzi, jízdu, létání atd. Ve virtuálním světě je možností navigace nespočet a jak se v prostoru pohybuje je důležité pro porozumění tohoto světa. Navigace je kombinace **hledání cesty** (*wayfinding*) a **cestování** (*travel*). Cestování popisuje, jak se uživatel pohybuje v prostoru. Hledání cesty vysvětluje, jak uživatel ví, kde se nachází a kam má namířeno [7].

3.6.2.1 Wayfinding

Hledání cesty (*wayfinding*) se týká uvědomění, kde se člověk nachází a nalezení cesty k cíli v daném prostoru. Hlavním krokem k dosažení tohoto cíle je vývoj kognitivní mapy nebo mentálního modelu prostředí, skrze které člověk prochází nebo plánuje. Znalost něčí polohy a okolí se nazývá **situační uvědomění** (*situational awareness*). Pokud nemáme situační uvědomění, jsme ztraceni. K orientaci v prostoru a nalezení cesty se používají **mentální modely** a pomůcky pro hledání cesty (*wayfinding aids*) [7].

Existují 4 metody pro tvorbu mentálního modelu. Metoda rozděle a panuj (*divide and conquer*) spočívá v rozdělení celkového prostředí na menší části a poté naučení cest mezi nimi. Během globální síťové strategie (*global network strategy*) si člověk zapamatovává snadno identifikovatelné lokace či mezníky. Postupné rozšiřování (*progressive expansion*) patří mezi méně úspěšné strategie, člověk se snaží zapamatovat část mapy a poté pokračuje dál. Poslední strategie a také ta nejméně úspěšná je narativní zpracování (*narrative elaboration*), tkví ve vytváření příběhů, během cesty, již člověk podnikl [7].

Pomůcek pro usnadnění navigace v prostoru existuje mnoho a jsou velkým pomocníkem k vytvoření mentálního modelu. Některé pomůcky jsou zabudované rovnou do prostředí a některé má člověk u sebe k dispozici. K pomůckám například patří mapy, mezníky, kompas, snadno zapamatovatelné názvy či zobrazení souřadnic [7].

3.6.2.2 Cestování

Pod pojmem **cestování** (*travel*) si člověk dokáže přestavit mnoho věcí, ale nejtypičtější je fyzický pohyb, např. chůze či běh. Dále existují i prostředky, které nám pomáhají se dostat z bodu A do bodu B rychleji, než je pro lidské tělo fyzicky možné. Létání v letadle či plavba na lodi je v dnešní době samozřejmostí a díky tomu mají vývojáři a designéři virtuální reality velké množství odkazů na různé druhy cestování, které se snaží implementovat do rozhraní, jež je nejvhodnější pro danou aplikaci. Lidé jsou schopni naučit se tato rozhraní, ale musí se dávat pozor, aby nebyla kontra intuitivní či těžká na pochopení. To by mohlo bránit uživatelům v zážitku nebo je to může i celkově odradit od používání. Dalším problémem, se kterým musí vývojáři a designéři počítat je, že lidé mají různé životní zkušenosti ovlivňující představu o tom, co je intuitivní. Jako příklad uvedu pilota a joystick – pokud si pilot bude přát klesat s letounem tak automaticky posune joystick dopředu. To však nemusí platit u uživatelů bez znalostí létání, tito uživatelé budou s největší pravděpodobností očekávat, že letoun začne po posunutí joysticku dopředu stoupat [7].

3.6.3 Interakce s ostatními

Interakce s ostatními ve virtuálním prostředí může také ovlivňovat náš zážitek. Pokud je hlavním účelem řešení problému a sdílení zážitku, poté mluvíme o spolupráci. Dále se můžeme setkat se soutěží, kde se uživatelé snaží nad ostatními vyhrát. Nebo může být hlavním cílem pouhá sociální interakce. Avšak většina virtuálních realit povoluje využití virtuálního prostředí v danou chvíli pouze jedním účastníkem. Většina těchto realit ovšem má možnost použití dalších monitorů, aby ostatní mohli pozorovat [7].

3.6.4 Tok interakce

Tok (*flow*) je faktor, který je přítomen ve všech typech interakce. Jedná se o dobu trvání interakce. Interakční tok může nastat dvěma způsoby: kontinuálně či diskrétně. Kontinuální tok znamená, že akce a/nebo reakce nastává průběžně v čase. V diskrétním toku nastává akce a/nebo reakce ihned. Z tohoto důvodu může uživatel vnímat interakci čtyřmi způsoby: kontinuální akce a kontinuální reakce (K–K), kontinuální akce a diskrétní reakce (K–D), diskrétní akce a kontinuální reakce (D–K) anebo diskrétní akce a diskrétní reakce (D–D) [8].

Příkladem (K–K) je například země v Google Earth (Obrázek 2), když se stiskne kurzor na mapě a pohybuje se s ním, model Země se podle toho otáčí.



Obrázek 2 – Google Earth [9]

V jedné studii se tyto druhy interakčních toků testovaly tím způsobem, že se vytvořily čtyři rozhraní. Každý druh interakčního toku byl reprezentován v jednom z rozhraní a vybrání uživatelé následně všechny odzkoušeli.

Velká část měla nejraději rozhraní s tokem K–K. Uživatelé měli rádi schopnost provádět nepřetržitou akci a získávat nepřetržitou reakci. Uživatelé takto cítili, že mají nad interakcí kontrolu. Na druhé straně interakční tok D–D byl nejméně oblíbený. Zjistilo se, že díky faktu, že nebyla akce ani reakce kontinuální, museli uživatelé více přemýšlet o tom, jak akci provádět a tím tak tyto akce chytřeji využívali. Díky tomu rychleji získali informace, které hledali a byli efektivnější než uživatelé s K–K. U toku K–D se uživatelům líbila možnost využití kontinuální akce a u D–K kontinuálnost reakce. Ze všech interakčních toků se na konci vyvodilo, že uživatelé preferovali kontinuálnost před diskrétností. Na druhou stranu diskrétní akce a reakce nutila uživatele hlouběji přemýšlet a tím problematiku rychleji pochopit [8].

3.7 Technologie

Tato podkapitola zmiňuje technologie, které jsou potřeba k vytvoření virtuální prohlídky. Stručně popisuje historii a důvody k výběru těchto technologií.

3.7.1 Unity

Unity je herní engine neboli software, který poskytuje vývojářům potřebné funkce k rychlé a efektivní tvorbě her. Herní engine je framework, který slučuje dohromady několik klíčových oblastí. Umožňuje nám importovat objekty z jiných softwarů a vkládat je do prostředí – přidávat světla, zvuky, speciální efekty, fyziku, animace, interaktivitu, herní logiku a následně upravovat, ladit a optimalizovat obsah na cílené platformy [10].

Unity bylo poprvé uvedeno na trh v roce 2005 s cílem vytvořit přístupnější proces vývoje her pro vývojáře. Nejaktuálnější verze je 2019.3.4. Unity je multiplatformní a umožňuje vytváření her na více než 25 různých platformách. Mezi těmito platformami je Android, iOS, Windows, Linux, Oculus Rift atd. Editor je podporován na Windows, macOS a Linux. V roce 2018 bylo Unity použito na vytvoření okolo poloviny her na mobilní telefony a 60 % aplikací na augmentovou a virtuální realitu [11, 12].

Důvodem výběru Unity je z velké části to, co ostatně bylo zmíněno v předchozím odstavci, je velmi populární pro vývoj mobilních aplikací. Také jsem v minulosti již s Unity pracoval a pár projektů vytvořil.

Základní stavební jednotkou v Unity je takzvaný **herní objekt** (*GameObject*). Herní objekty reprezentují všechny části v aplikaci, jako je například hráč, krajina či světlo. Tyto objekty však samy od sebe nic nedělají, a proto je nutné jim přidat určité vlastnosti. Aby nabily těchto vlastností musí mít na sobě přiřazené komponenty. V závislosti na typu objektu, který chcete vytvořit, přidáte na objekty různé kombinace těchto komponentů. Unity má v sobě již mnoho různých předvytvořených komponent, které se dají využít, ale uživatel si může vytvořit i vlastní pomocí skriptování [13].

Skripty v Unity jsou kusy kódu, které umožňují vytvořit si vlastní komponenty, spouštět herní události, upravovat vlastnosti komponenty a reagovat na vstup uživatele. Převážná většina těchto skriptů je napsána v jazyce C# [14].

3.7.2 Android

Android je mobilní operační systém vytvořen společností Android. Inc., která byla odkoupena společností Google v roce 2007. Aktuální verze je Android 10 vydaná 3. září 2019. Android je od roku 2011 nejprodávanější operační systém pro mobilní telefony

a od roku 2013 i pro tablety. Má přes 2,5 miliardy aktivních uživatelů a na začátku roku 2020 bylo na Google Play Store přes 2,8 milionu aplikací [15, 16].

Hlavním argumentem k využití mobilním operačního systému Android oproti jiným je převážně to, že je nejpoužívanější. Dále také proto, že sám vlastním mobilní telefon s operačním systémem Android a mohu ho tedy využívat k testování vyvíjené virtuální prohlídky.

4 Vlastní práce

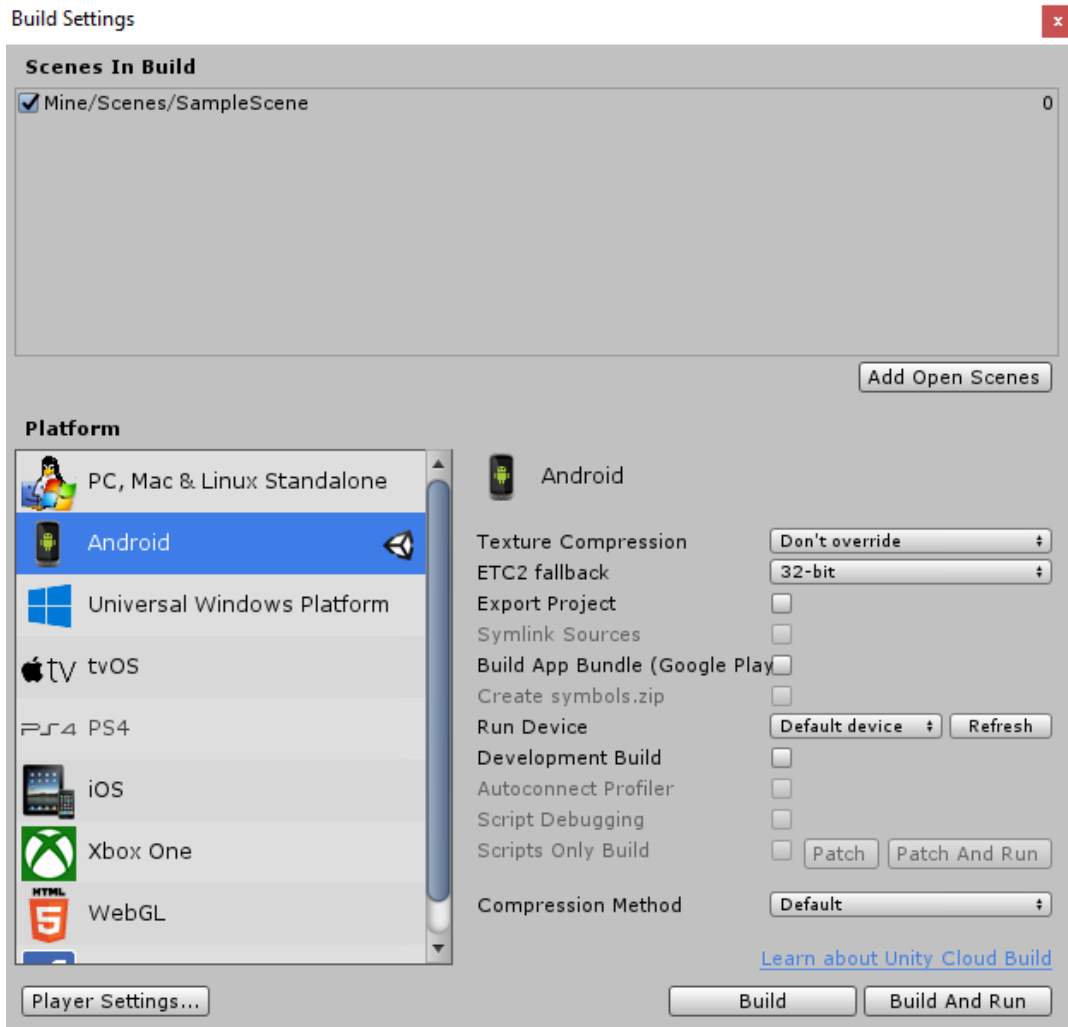
Praktická část se věnuje tvorbě virtuální prohlídky pro operační systém Android. Vývoj prohlídky je popsán od začátku pro obchod s hudebninami, který bude vymodelován. Samotný proces modelování prodejny není popisován. Hlavním cílem je vytvořit virtuální prostředí, ve kterém se bude uživatel moci pohybovat a interagovat s objekty, které se v něm nacházejí. Interakce bude možná jen pohledem, a to z toho důvodu, že ne všechny brýle na trhu, do kterých se vkládá mobil mají tlačítka k dispozici. Proto je závěrem úvaha, zdali je interakce pohledem k aplikaci tohoto druhu dostatečná.

V dokončené aplikaci si bude uživatel schopen prohlížet hudební nástroje, které budou v místnosti kolem rozestaveny. Pohyb bude zprostředkován tak, že budou malé modré objekty rozestaveny na podlaze a pokud se na ně hráč podívá, teleportuje se na jejich pozici. Pokud bude uživatel chtít zjistit více informací o hudebním nástroji, má možnost na daný nástroj upřít zrak a po krátké době bude přemístěn na jiné místo, kde se mu zobrazí informace jako je název a cena. Dále se kousek od uživatele objeví otáčející kopie nástroje, který si vybral, aby měl možnost si jej detailněji prohlédnout. Pod informacemi bude tlačítko, jež uživatele vrátí zpět na místo odkud byl teleportován. Nakonec bude k dispozici třídění, podle kterého bude uživatel mít možnost vyfiltrovat nástroje, jež ho zajímají.

4.1 Příprava

V Unity se vytvoří nový 3D projekt, který se pojmenuje a uloží. Dále se stáhne Google VR SDK (sada vývojových nástrojů) pro Unity vytvořené společností Google, která vývojářům usnadňuje rychlejší a snadnější tvorbu aplikací na virtuální realitu v Unity.

Důvodem využití VR SDK je především to, že obsahuje základní funkcionalitu pro VR aplikace v Unity pro Android, čímž pomáhá vývojářům urychlit vývoj a také zajistit optimálního fungování základních mechanik pro tyto aplikace. Vývojáři si tak můžou být jistí, že elementární prvky, jako je např. sledování pohybu hlavy, jsou optimálně naimplementované a vývojáři mohou pracovat na specifických problémech. Po stažení se SDK importuje do projektu. Dále se přepne v nastavení platforma na Android (Obrázek 3), aby bylo možné testovat pro mobilní zařízení a klikne se na tlačítko *Add Open Scenes*.



Obrázek 3 – Nastavení pro spuštění aplikace na platformě Android

4.2 Pohyb

Do počáteční scény je nutné přidat 3 objekty z importovaného VR SDK. Prvním je *GvrEditorEmulator*, který umožňuje emulovat pohyb hlavy přes tlačítka na klávesnici a tím značně ulehčit testování bez opakované instalace testované verze na mobilní telefon. Druhý objekt je *GvrReticlePointer*. Ten se vloží na hlavní kameru, která reprezentuje obraz, jenž vidí uživatel a přidá do jejího centra ukazatel, jenž bude zprostředkovávat interakci s okolím. A nakonec *GvrEventSystem*, který přidává možnost interakce mezi ukazatelem v centru obrazovky a objekty v prostoru.

Dalším úkolem je přidat uživateli schopnost pohybovat se v prostoru. Jelikož je k dispozici jen pohled, tak se jeví jako nejlepší způsob přidat do prostoru objekty, které změni polohu uživatele tehdy, když se na ně podívá. Uživatel se tak objeví na stejné pozici

jako dané objekty. Objekty, které se na tuto funkci budou využívat, musí mít na sobě spouštěče, jež umožní spustit skript pro změnu polohy uživatele (Obrázek 5). Spouštěč se objektu přidá pomocí *Event Trigger* z importovaného SDK. V inspektoru objektu se klikne na *Add Component* a najde se *Event Trigger*. *Event trigger* je ze začátku prázdný. Musí se přidat typ spouštěče (*event type*). Na výběr jich zde je mnoho. V této situaci se použije *PointerEnter*. To je spouštěč, který se spustí při události, kdy ukazatel vizuálně vstoupí do objektu s tímto spouštěčem. Jelikož neexistuje ještě skript, který by se spustil, nebude se nic dít. V následujícím kroku tedy vytvořím potřebný skript.

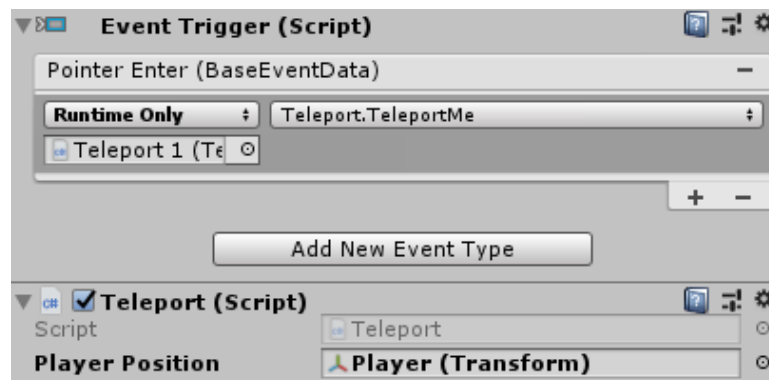
Skript pracuje s objektem *playerPosition*, který reprezentuje hráče, a objektem *teleportPosition*, který představuje místo, kam se bude hráč teleportovat. Skript se přidá jako komponent na teleport a objekt hráče se vloží do kolonky *Player Position* aby skript věděl co má považovat za objekt hráče. Nakonec se přidá komponent *Event Trigger* (Obrázek 4), a nastaví se tak aby se funkce *TeleportMe()* vyvolala, poté co ukazatel vstoupí to objektu pro teleport.

```
public class Teleport : MonoBehaviour
{
    public Transform playerPosition;

    private int playersYCoord = 1;
    private Transform teleportPosition;

    // Start is called before the first frame update
    void Start()
    {
        teleportPosition = GetComponent<Transform>();
    }

    public void TeleportMe()
    {
        playerPosition.position = new Vector3(teleportPosition.position.x,
        playersYCoord, teleportPosition.position.z);
    }
}
```



Obrázek 4 – Teleport skript jako komponenta

Takto nastavené objekty se přidají do scény a uživatel má nyní možnost se v prostředí přemísťovat.



Obrázek 5 – Teleportační objekty v aplikaci

4.3 Indikátor průběhu

Tato virtuální prohlídka bude mít interaktivní prvky, než se tak stane, musí být nalezen způsob, jak se bude moci pouhým pohledem s objekty v prostoru interagovat. Nejlepší a nejintuitivnější řešení je naimplementovat indikátor průběhu, jenž se bude postupně načítat v případě, kdy ukazatel na hlavní kameře vstoupí a zůstane na objektu, se kterým uživatel bude chtít interagovat.

Jako indikátor pro průběh byl vybrán červený kruh (viz Obrázek 6). Ve skriptu nastavíme indikátor tak, aby se postupem času vyplňoval v okamžiku, kdy hráč kouká ukazatelem na kameře na zvolený objekt.

Proměnná *totalTime* udává, kolik sekund musí uběhnout, aby se indikátor zcela vyplnil a *timer* si ukládá čas, který zatím od počátku odečítání uběhl. Booleovská proměnná *status* si ukládá informaci, zdali se hráč s objektem snaží interagovat a míří ukazatelem na objekt či nikoliv. Podle stavu této proměnné skript ví, jestli se má indikátor průběhu načítat.



Obrázek 6 – Indikátor průběhu

```

public class ObjectInteraction : MonoBehaviour
{
    public Image loadingImage;
    public float totalTime = 2;
    bool status;
    public float timer;

    // Update is called once per frame
    void Update()
    {
        if(status)
        {
            timer += Time.deltaTime;
            loadingImage.fillAmount = timer / totalTime;
        }

        if (timer > totalTime)
        {
            timer = 0;
        }
    }
}

```

Dále se vytvoří dvě funkce. Funkce *Activated()* se spustí pokud hráč vstoupí s ukazatelem do objektu a tím nastaví *status* na pravdivou hodnotu a aktivuje načítání indikátoru. Funkce *Deactivation()* se na druhou stranu spustí, když ukazatel objekt opustí a tím změní *status* na nepravdivou hodnotu a resetuje načítání indikátoru.

```

public class ObjectInteraction : MonoBehaviour
{
    public void Activated()
    {
        status = true;
    }

    public void Deactivated()
    {
        status = false;
        timer = 0;
        loadingImage.fillAmount = 0;
    }
}

```

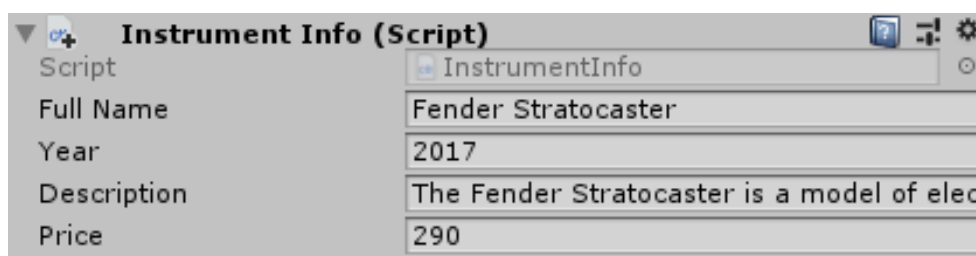
4.4 Informace o nástroji

Ve finální verzi bude uživatel mít možnost zjistit podrobnější informace o konkrétním nástroji. Proto je nutné vytvořit plátno (*canvas*), na kterém se informace o hudebním nástroji budou vyobrazovat. V Unity je vytvoření takového plátna jednoduché. Do scény se přidá objekt *canvas* a na ten je poté možné přidat textová pole, do kterých se budou jednotlivé informace zobrazovat.

Pro každý nástroj bude existovat informace o jeho celém názvu, roku výroby, ceně, a nakonec i jeho podrobnější popis. O držení těchto informací se bude starat třída *InstrumentInfo* (Obrázek 7). Skript s touto třídou bude přidán, ke každému z existujících nástrojů.

```
public class InstrumentInfo : MonoBehaviour
{
    public string fullName = "Instrument Name";
    public string year = "Year";
    public string description = "Instrument Description";
    public int price = 0;
}
```

Přidáním skriptu ke každému z nástrojů umožní informace o nástroji vložit do kolonek.



Obrázek 7 – *InstrumentInfo* skript jako komponenta

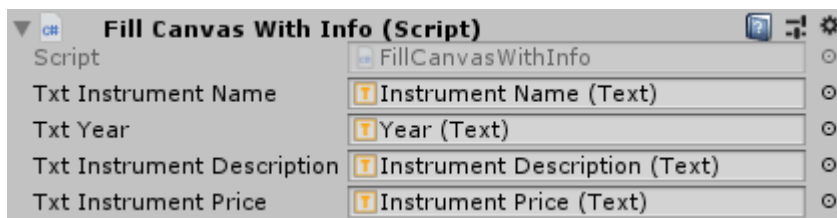
4.5 Příprava plátna pro zobrazení informací

Po vytvoření plátna s textovými poli se musí vytvořit skript, který se na textová pole bude odkazovat, aby aplikace věděla, kam informace o nástrojích vložit. Pro tento účel byl vytvořen skript *FillCanvasWithInfo*. Tento skript bude mít k dispozici odkazy na textová pole. Funkce *FillInformation()* přijímá 4 argumenty a to jméno instrumentu (*name*), rok výroby (*year*) podrobnější popis (*description*) a cenu nástroje (*price*) (viz Obrázek 8). Obsah těchto argumentů se následně přiřadí do jednotlivých textových polí na plátně.

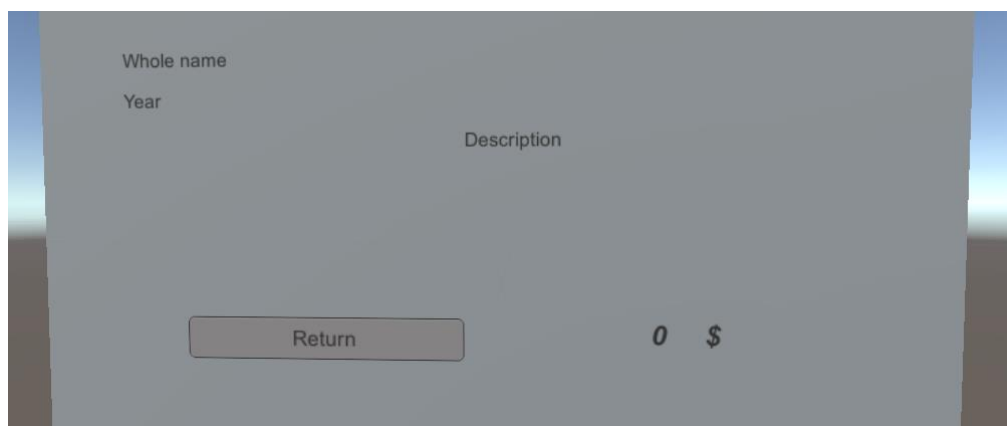
```
public class FillCanvasWithInfo : MonoBehaviour
{
    public Text txtInstrumentName;
    public Text txtYear;
    public Text txtInstrumentDescription;
    public Text txtInstrumentPrice;

    public void FillInformation(string name, string year, string description, int price)
    {
        txtInstrumentName.text = name;
        txtYear.text = year;
        txtInstrumentDescription.text = description;
        txtInstrumentPrice.text = price.ToString();
    }
}
```

Skript se přidá jako komponenta k plátnu a manuálně se vloží reference k textovým polím.



Obrázek 8 – *FillCanvasWithInfo* skript jako komponenta



Obrázek 9 – Nevyplněné plátno

4.6 Teleport k plátnu a vytvoření kopie nástroje

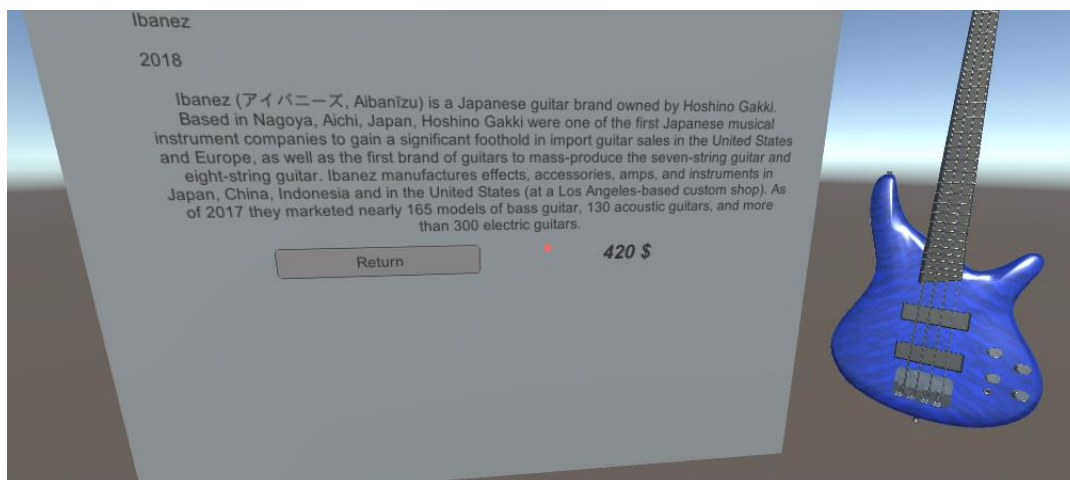
V této části se vytvoří skript, který teleportuje hráče na místo, kde se nachází plátno s informacemi (Obrázek 10). U plátna se souběžně vytvoří otáčející se kopie nástroje, ke kterému se informace vztahují. Dále se napíše kód na teleportaci zpět a s tím i smazání kopie nástroje pro uvolnění paměti.

Plátno s informacemi bude umístěno na jiné místo než, kde se uživatel vyskytuje nyní, a proto je potřeba uživatele k plátnu teleportovat. Jelikož se bude daný uživatel teleportovat zpět potřebujeme si jeho pozici uložit. S tím si uložíme i název nástroje, ke kterému vytvoříme kopii. Tento název později využijeme při mazání kopie.

```
public class PlayerTransform : MonoBehaviour
{
    public Vector3 Position { get; set; }
    public string InstrumentName { get; set; }
}
```

V dalším kroku se nebude vytvářet nový skript, ale se přidá kód do již existujícího skriptu *ObjectInteraction*. Jako první krok se vyplní plátno pomocí funkce *FillInformation*.

```
fillCanvas.FillInformation(info.fullName, info.year, info.description,
    info.price);
```



Obrázek 10 – Vyplněné plátno s informacemi a kopie hudebního nástroje

Následně se před teleportací uloží pozice uživatele a název nástroje, se kterým interaguje. V dalším kroku se uživatel teleportuje. Vytvořením instance třídy *Vector3* a přiřazením k uživateli se přepsali souřadnice na jiné, což se uživateli v aplikaci jeví jako by byl teleportován. Poté se funkcí *Instantiate* vytvoří kopie nástroje. Do prvního argumentu této funkce se vloží objekt a souřadnice, na kterých se objeví.

```
player.Position = playerPosition.position;
player.InstrumentName = gameObject.name;
```

```
playerPosition.position = new Vector3(canvasInfoPosition.position.x,
    canvasInfoPosition.position.y, canvasInfoPosition.position.z-1);
```

```
GameObject clone = Instantiate(gameObject, new Vector3(playerPosition.position.x+2,
    playerPosition.position.y, playerPosition.position.z), Quaternion.identity);
```

Tato kopie se bude otáčet, tak aby si ji mohl uživatel prohlédnout ze všech stran. Pro tento účel se vytvoří skript *RotateObject*.

```
public class RotateObject : MonoBehaviour
{
    void Start()
    {
        transform.rotation = Quaternion.Euler(70, 0, 0);
    }

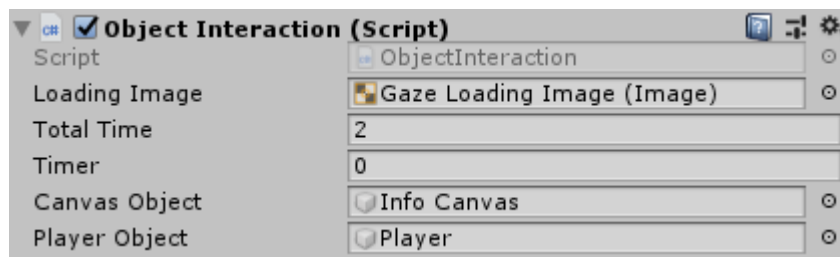
    // Update is called once per frame
    void Update()
    {
        transform.Rotate(new Vector3(0, 1, 0), Space.World);
    }
}
```

Pomocí funkce *AddComponent* se tento skript přidá ke kopii. Na závěr se deaktivuje na kopii skript *ObjectInteraction* jelikož nechceme aby bylo možné z touto kopií interagovat jako s originálním objektem.

```
clone.AddComponent<RotateObject>();  
clone.GetComponent<ObjectInteraction>().enabled = false;
```

V případě teleportování uživatele zpět a zničení kopie je kód kratší. Kopie se nejprve nalezne pomocí názvu, který se uložil a funkce *Find()*, která objekt podle názvu najde. Nalezený objekt se poté funkcí *Destroy()* zničí. Na konec se uživatel teleportuje zpět na pozici, která se předem také uložila.

```
else  
{  
    Destroy(GameObject.Find(player.InstrumentName + "(Clone)"));  
    playerPosition.position = player.Position;  
}
```



Obrázek 11 – *ObjectInteraction* skript jako komponenta

4.7 Třídění

V aplikaci bude možnost vytřídit nástroje podle kritérií. Jelikož je uživatel omezen jen na používání pohledu, bude se třídění provádět pouze z dostupných přednastavených možností. Ve hře bude vytvořené plátno s tlačítky, na kterých bude napsaná podmínka, jež se provede poté, co ji uživatel pohledem vybere. Nástroje, které nebudou podmínce vyhovovat, zmizí, do té doby, než je uživatel neresetuje či nezmění podmínku.

Aby aplikace zaregistrovala, že bylo tlačítko zmáčknuté, musí se vyvolat funkce *Invoke()*, která poté provede akci na třídění.

```
if (timer > totalTime)  
{  
    timer = 0;  
    buttonClick.Invoke();  
}
```

Dostupné je třídění podle druhu nástroje, tj. kytara nebo basa, a také podle ceny, tj. zda je nástroj levnější než 300 nebo 500 dolarů (viz Obrázek 12).



Obrázek 12 – Možnosti třídění

Pro filtrování existuje skript *FilterInstruments*, který obsahuje dvě privátní funkce s argumenty. První z těchto funkcí je *ShowByType()*, která má jako argument řetězec. Poté prohledává všechny hudební nástroje, ty, které mají v názvu tento řetězec nastaví jako viditelné a ostatní deaktivuje, což odstraní jejich modely ze hry.

```
private void ShowByType(string typeOfInstrument)
{
    foreach (var child in children)
    {
        if (child.name.Contains(typeOfInstrument) || child.name.Contains("All"))
        {
            child.gameObject.SetActive(true);
        }
        else
        {
            child.gameObject.SetActive(false);
        }
    }
}
```

Druhá funkce *ShowByPrice()* je podobná té první, ale rozdíl je v tom, že jako argument má celé číslo. Toto číslo reprezentuje hodnotu nástroje a pokud v množině všech nástrojů najde nástroje s nižší cenou, aktivuje je. Ostatní deaktivuje, čímž se provede třídění.


```

private void ShowByPrice(int price)
{
    foreach (var child in children)
    {
        InstrumentInfo info = child.gameObject.GetComponent("InstrumentInfo") as
        InstrumentInfo;
        if(info)
        {
            if (info.price < price)
            {
                child.gameObject.SetActive(true);
            }
            else
            {
                child.gameObject.SetActive(false);
            }
        }
    }
}

```

Tyto dvě funkce jsou dále použity jako základ pro podmínky třídění. Pokud bude do funkce *ShowByType()* vloženo slovo *guitar* jako argument, funkce najde v aplikaci jen kytary a ostatní nástroje deaktivuje. To samé platí pro funkci *ShowByPrice()*, pro celočíselný argument reprezentující cenu v dolarech.

Pokud bude chtít uživatel vytrždit jen kytary, vyvolá se funkce *ShowByType()*.

```

public void ShowBasses()
{
    ShowByType("Guitar");
}

```

Pokud bude chtít uživatel vytrždit jen nástroje, které stojí méně než 300 dolarů, vyvolá se funkce *ShowByPrice()*.

```

public void PriceLessThan300()
{
    ShowByPrice(300);
}

```

Pro resetování třídění nemusíme vytvářet novou funkci, stačí spustit funkci *ShowByType()* s prázdným argumentem. Jelikož prázdnému argumentu vyhovují všechny nástroje, třídění se vyresetuje a objeví se všechny nástroje.

```

public void Reset()
{
    ShowByType("");
}

```


5 Výsledky

Před začátkem vytváření aplikace byla do projektu přidána sada vývojových nástrojů od společnosti Google. Tato sada velmi ulehčila implementaci základních prvků této aplikace. Především objekty *GvrEditorEmulator*, *GvrReticlePointer* a *GvrEventSystem*, se kterými bylo rychlé nastavit základní funkcionalitu a mohlo se rychle přejít na specifitější problémy, které byly potřebné naimplementovat.

Aplikace byla od počátku tvořena tak, že nebude možnost používat žádná tlačítka či jiná externí zařízení, ale pouze pohyb hlavy a pohled. V této fázi je již aplikace hotová, proto mohu usoudit, že takto navržené ovládání této aplikace je dostatečné. V aplikaci je vše naimplementováno a nenastal problém, který by nebyl řešitelný pohybem hlavy či pohledem. Nicméně použití tlačítka či externího zařízení by některé funkce zrychlilo a samotné ovládání by bylo uživatelsky pohodlnější.

Objekty hudebních nástrojů jsou do aplikace vkládány po jednom manuálně. Tento fakt, je důvodem, proč je ve finální aplikaci pouze omezený počet hudebních nástrojů. V budoucí verzi této aplikace by bylo výhodné vkládání nástrojů ulehčit a tím i zrychlit.

Jedním z větších nedostatků jsou modely použité pro zobrazení hudebních nástrojů. Na mém mobilním zařízení Xiaomi Redmi Note 4 má aplikace nízký počet snímků za sekundu, tím se zpomaluje a negativně narušuje pocit z používání. Na výkonnějších zařízeních, jako je například Redmi Note 7 a mobilní zařízení s podobným hardwarem, problém není.

6 Závěr

Cílem práce je seznámení s problematikou interaktivní virtuální prohlídky pro mobilní telefon a tvorba interaktivní virtuální prohlídky. Byla probrána potřebná terminologie k pochopení základů virtuální reality a interaktivity, a také používané technologie pro vytvoření aplikace. Virtuální prohlídka byla vytvořena pro obchod s hudebninami. Aplikace byla vytvořena v Unity pro mobilní telefon s operačním systémem Android.

Do prohlídky byla přidána možnost interakce pohledem, respektive namířením ukazatele ve středu obrazovky na objekt zájmu po dobu naplnění indikátoru spuštění interakce. Tato aplikace může být vhodná pro budoucí vývojáře, kteří budou chtít přidat interaktivní prvky do své virtuální prohlídky.

Funkcionalita, která by se mohla přidat do rozšířené verze, je možnost porovnání zvolených hudebních nástrojů. V aktuální verzi je zatím možné prohlédnout si nástroje jen jednotlivě.

7 Seznam zdrojů

1. WEINBAUM, Stanley G. Pygmalion's Spectacles. Montana, USA: Kessinger Publishing, 2010. ISBN 978-1169173781.
2. History of VR – Timeline of Events and Tech Development. VirtualSpeech – Soft Skills Training with VR [online]. Dostupné z: <https://virtualspeech.com/blog/history-of-vr>
3. Oculus VR. In: https://en.wikipedia.org/wiki/Oculus_VR [online]. [cit. 2020-01-21]. Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/thumb/0/09/Oculus-Rift-CV1-Headset-Front.jpg/1280px-Oculus-Rift-CV1-Headset-Front.jpg>
4. LAVALLE, Steven M. VIRTUAL REALITY [online]. Cambridge University, 2019 [cit. 2019-09-04]. Dostupné z: <http://vr.cs.uiuc.edu/vrbooka4.pdf>. Skripta. University of Oulu.
5. KUKSA, Iryna a Mark CHILDS. Making sense of space: the design and experience of virtual spaces as a tool for communication. Oxford, UK: Chandos Publishing, an imprint of Elsevier Limited, 2014. ISBN 978-184-3347-408.
6. JENSEN, Jens Fog. Interactivity: Tracking a New Concept in Media and Communication Studies. 1998. Dostupné z: <https://www.semanticscholar.org/paper/Interactivity%3A-Tracking-a-New-Concept-in-Media-and-Jensen/b04de1353d87619b40102fcab04f393615ba4e10>
7. SHERMAN, William R. a Alan B. CRAIG. Understanding virtual reality: interface, application, and design. Second edition. Cambridge, MA: Morgan Kaufmann, [2019]. Morgan Kaufmann series in computer graphics. ISBN 978-012-8010-389.
8. Hai-Ning LIANG, Paul PARSONS, Hsien-Chi WU a Kamran SEDIG. An Exploratory Study of Interactivity in Visualization Tools: "Flow" of Interaction. 2010. Dostupné z: https://www.researchgate.net/publication/234609890_An_Exploratory_Study_of_Interactivity_in_Visualization_Tools_Flow_of_Interaction
9. Google Earth [online]. [cit. 2020-03-05]. Dostupné z: <https://www.google.com/earth/>
10. Game engines—how do they work? [online]. [cit. 2020-03-05]. Dostupné z: <https://unity3d.com/what-is-a-game-engine>
11. Unity download archive [online]. [cit. 2020-03-05]. Dostupné z: <https://unity3d.com/get-unity/download/archive>

12. Unity – Our Company [online]. [cit. 2020-03-05]. Dostupné z: <https://unity.com/our-company>
13. GameObjects [online]. [cit. 2020-03-05]. Dostupné z: <https://docs.unity3d.com/Manual/GameObjects.html>
14. Creating and Using Scripts [online]. [cit. 2020-03-05]. Dostupné z: <https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>
15. Active Android devices [online]. [cit. 2020-03-05]. Dostupné z: <https://www.androidpolice.com/2019/05/07/there-are-now-more-than-2-5-billion-active-android-devices/>
16. Number of Android apps on Google Play [online]. [cit. 2020-03-05]. Dostupné z: <https://www.appbrain.com/stats/number-of-android-apps>

8 Přílohy

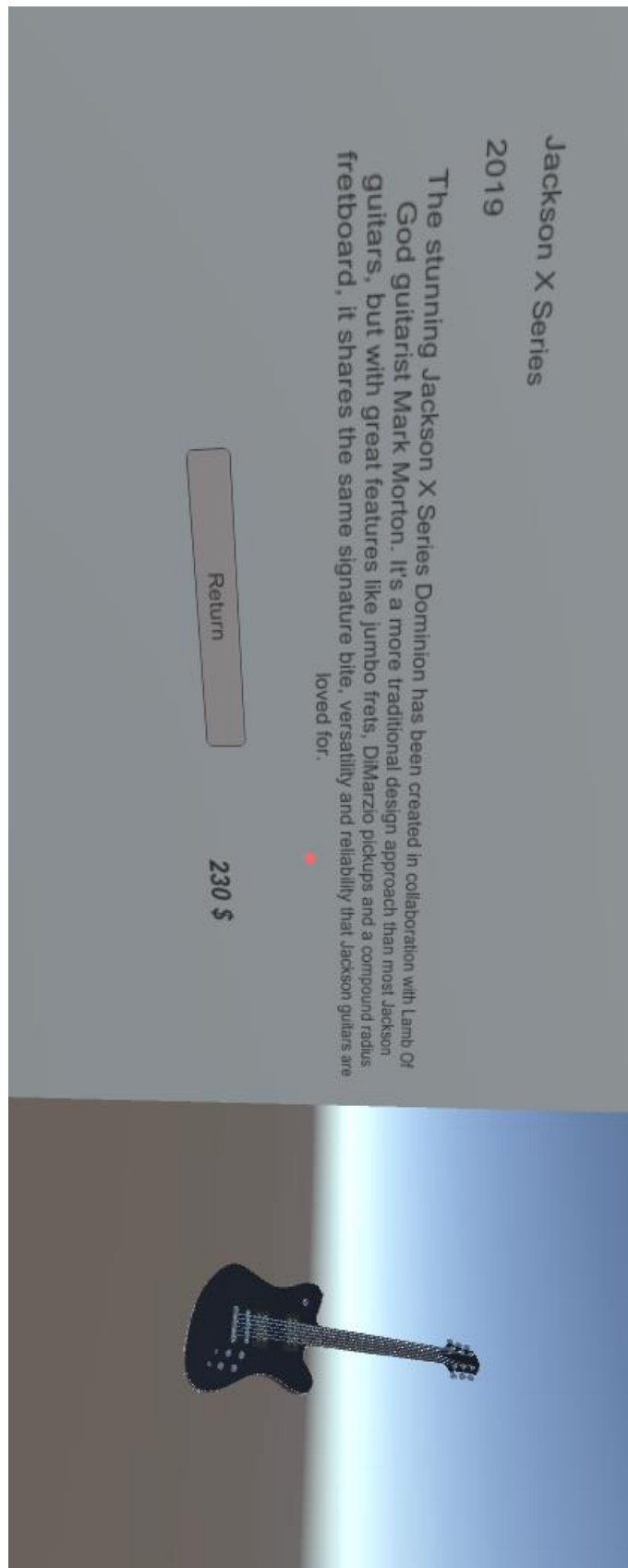
8.1 Instalační aplikační balíček a vytvořené skripty

Přílohou této bakalářské práce je CD s vytvořenými skripty a aplikační balíček, pomocí kterého je možné aplikaci na mobilní telefon nainstalovat.

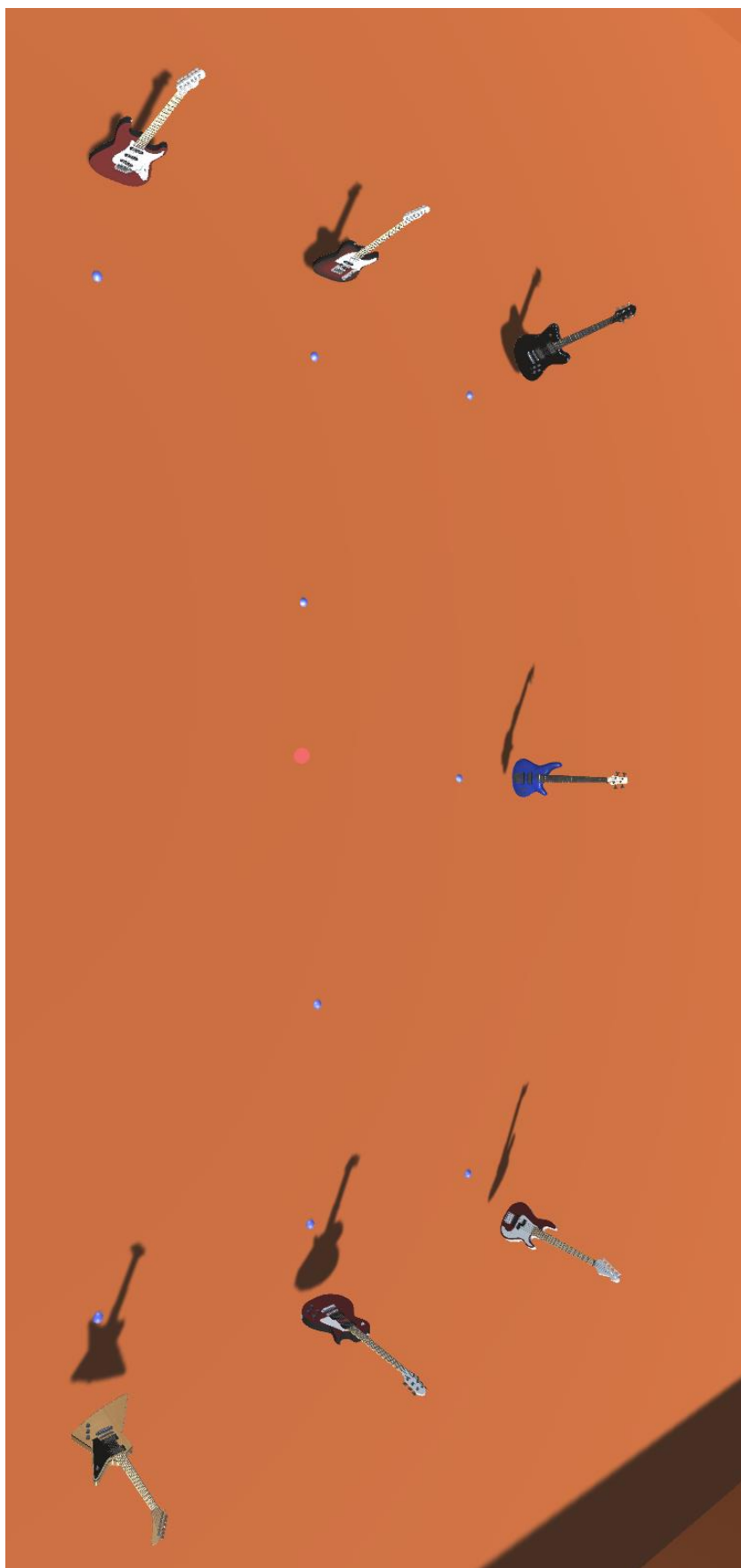
8.2 Snímky z aplikace



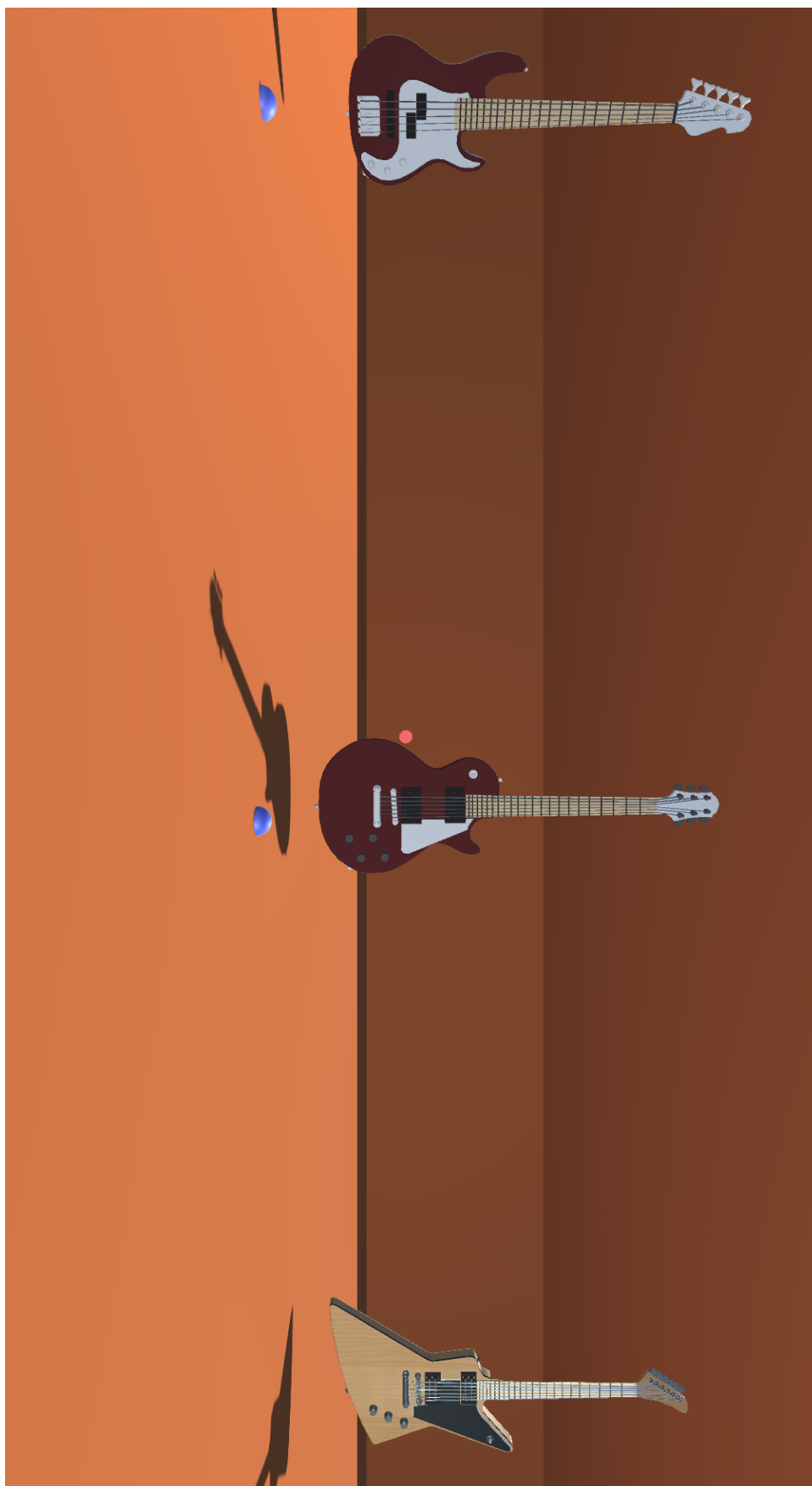
Obrázek 15 Rotující hudební nástroj



Obrázek 16 Plátno s informacemi o vybraném hudebním nástroji s rotující kopíí daného nástroje



Obrázek 17 Hudební nástroje z výšky



Obrázek 18 Hudební nástroje z pohledu uživatele