Vyhodnocení parametrů AR kódů pro využití v Rozšířené Realitě

Adam Kopřiva

Bakalářská práce
2024

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Adam Kopřiva**
Osobní číslo:	**A21058**
Studijní program:	**B0613A140020 Softwarové inženýrství**
Forma studia:	**Prezenční**
Téma práce:	**Vyhodnocení parametrů AR kódů pro využití v Rozšířené Realitě**
Téma práce anglicky:	**Evaluation of AR Code Parameters for Use in Augmented Reality**

Zásady pro vypracování

1. Sepište aktuální typy kódů Rozšířené Reality (Augumented Reality, AR) pro rozšířenou realitu a vyhodnoťte jejich vhodnost pro různé typy aplikací.
2. Popište princip a vytvoření rozšířené reality s využitím kódů Rozšířené Reality (Augumented Reality, AR) v programu Unreal Engine.
3. Navrhněte a vytvořte vlastní kód Rozšířené Reality (Augumented Reality, AR) pro skenování a vkládání dodatečného obsahu pomocí rozšířené reality.
4. Otestujte a porovnejte vytvořený kód Rozšířené Reality (Augumented Reality, AR) s již existujícími kódy Rozšířené Reality (Augumented Reality, AR).

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. ONG, Soh Khim, YEH CHING NEE, Andrew, ed., 2023. *Handbook of Augmented Reality*. Springer. ISBN 978-3-030--67821-0.
2. *Unreal Engine 4 Documentation* [online], 2023. [cit. 2023-12-08]. Dostupné z: https://docs.unrealengine.com/4.27/en-US/
3. COOKSON, Aram, Ryan DOWLINGSOKA a Clinton CRUMPLER, 2016. *Sams Teach Yourself Unreal® Engine 4 Game Development in 24 Hours*. Pearson Education. ISBN 978-0-672-33762-8.
4. VALCASARA, Nicola, 2015. *Unreal Engine Game Development Blueprints*. Packt Publishing. ISBN 978-1-78439-777-7.
5. NIXON, David, 2020. *Beginning Unreal Game Development*. Apress. ISBN 978-1-4842-5638-1.

Vedoucí bakalářské práce: **Ing. Václav Mach, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **5. listopadu 2023**
Termín odevzdání bakalářské práce: **13. května 2024**

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
DĚKAN
L.S.

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 5. ledna 2024

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Adam Kopřiva, v.r.

Ve Zlíně, dne 13.5.2024 …….

podpis studenta

ABSTRAKT

Cílem práce je sepsání aktuálních AR kódu pro rozšířenou realitu a vyhodnotit jejich vhodnost pro různé typy aplikací. Testování jednotlivých AR kódů bude probíhat pomocí mobilní aplikace s podporou operačního systému Android, která bude vyvíjena v programu Unreal Engine, který obsahuje všechny podstatné moduly pro vývoj dané aplikace. Na základě testování již existujících AR kódů bude dalším cílem vytvoření vlastního typu AR kódu, který bude vhodný pro skenování a následné vkládání dodatečného obsahu pomocí rozšířené reality.

Klíčová slova: Rozšířená Realita, Unreal Engine, Android, AR kód, Testování, Blueprint

ABSTRACT

The aim of this work is to write actual AR code for augmented reality and evaluate its suitability for different types of applications. The testing of each AR code will be done using a mobile application with support for the Android operating system, which will be developed in the Unreal Engine, which contains all the essential modules for the development of the application. Based on the testing of existing AR codes, the next goal will be to create a custom type of AR code that will be suitable for scanning and then inserting additional content using augmented reality.

Keywords: Augumented Reality, Unreal Engine, Android, AR code, Testing, Blueprint

Na tomto místě bych chtěl poděkovat vedoucímu práce Ing. Václavu Machovi, Ph.D., za cenné rady a trpělivost.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD

Rozšířená realita (AR) představuje technologický koncept spojující reálný svět s digitálními prvky, čímž obohacuje uživatelský zážitek interakce s okolním prostředím. Tato technologie umožňuje vytvářet virtuální svět, transformuje naše vnímání a interakci s prostředím kolem nás. V první části práce jsou systematicky popsány a analyzovány dostupné kódy AR, s důrazem na jejich aktuální využitelnost a potenciální aplikace. Cílem této části je poskytnout ucelený přehled o stávajících technologiích a identifikovat trendy v oblasti rozšířené reality.

Druhá část bakalářské práce se zaměřuje na principy a vytvoření rozšířené reality s využitím kódů AR v prostředí programu Unreal Engine. Tato část poskytne podrobný návod na implementaci AR do tohoto herního enginu, včetně klíčových kroků a postupů. Cílem této části je umožnit čtenáři prakticky vstoupit do procesu vývoje AR aplikací pomocí konkrétního nástroje.

Praktická část práce je věnována návrhu a implementaci vlastního kódu AR pro skenování a vkládání dodatečného obsahu prostřednictvím rozšířené reality. Zde jsou detailně popsány použité metody a techniky, a jsou provedeny demonstrace vytvořeného kódu na praktickém příkladu.

Poslední část práce se zaměřuje na testování a porovnání vytvořeného kódu AR s již existujícími kódy Rozšířené Reality. Byly provedeny konkrétní testovací scénáře, a následně jsou analyzovány výsledky. Cílem této části je poskytnout ucelený pohled na efektivitu a možnosti nově navrženého kódu v porovnání s existujícími standardy v oblasti rozšířené reality. Celkovým záměrem bakalářské práce je tedy přispět k dalšímu rozvoji a porozumění oblasti AR kódů, a zároveň poskytnout konkrétní praktický přínos pro vývojáře a uživatele v této oblasti.

I. TEORETICKÁ ČÁST

1 ROZŠÍŘENÁ REALITA

Rozšířená realita umožnuje vkládat v reálném světě a čase 3D objekty, animace, zvuky pomocí simulovaných hologramů do mobilních zařízení, chytrých brýlí a headsetů. Existuje zde několik pojmů, které se často zaměňují s pojmem rozšířená realita, proto je dobré tyto pojmy od sebe odlišit. [1]

1.1.1 Virtuální realita

Obvykle se jedná o uzavřené headset zařízení, ve kterém se vytváří 3D svět vytvořený počítačem, ve kterém se uživatel může pohybovat a interagovat pomocí příslušenství od určitého výrobce daného headsetu. Například se jedná o všesměrové běžecké pásy simulující chůzi nebo běh, speciálně upravené rukavice, které se mohou využít u first person shooting her. [2]

1.1.2 Hybridní realita

Hybridní realita se snaží sloučit skutečný a virtuální svět, a to pomocí tří podstatných prvků. Mezi tyto prvky patří využití výpočetní síly cloudových řešení, skutečná realita a vstřebávání okolního prostředí. Tyto prvky se navzájem propojují, toto propojení se odborně nazývá jako spektrum hybridní reality, tedy že zároveň existuje skutečná realita, která se prolíná s realitou uměle vytvořenou. Existují základní dva druhy zařízení Windows Mixed Reality a to Holografické zařízení, které do skutečné reality dokáže dodat virtuální obsah. Zatímco Asistivní zařízení se odlišují tím, že fyzickou realitu zcela přeměňují na realitu virtuální. [3]

1.2 Historie rozšířené reality

V roce 1901 spisovatel Frank L. Baum vydal dílo The Master Key, které obsahovalo první náznak rozšířené reality se speciálně upravenými brýlemi, pomocí kterých se dalo spatřit písmeno na hlavách daných postav, které označovalo nějakou vlastnost charakteru dané postavy, například písmeno C označovalo krutost nebo písmeno E značilo, že je postava zlého zdání. První pokusy o vytvoření skutečné rozšířené reality nastaly až o 56 let později, kdy kinematograf Mortan Heilig měl vizi mnohoúrovňového promítacího sálu, který by dokázal zároveň zobrazit obraz, zvuk s vibracemi a vůni přímo v jedné místnosti. V roce 1962 mu byl udělen patent na první prototyp zařízení, které se jmenovalo Sensorama Machine, které umožnovalo projekci 3D filmu se zvukem, imitaci větru ve vlasech a vibrace s vůní, která se aktivovala v určitý časový okamžik. Tento prototyp byl funkční, ale nenašel

ve své době finanční podporu, aby se mohl v plné výši rozvinout a dotáhnout do konce. Pojem rozšířená realita se poprvé objevil ve devadesátých letech minulého století, kdy dva badatelé Thomas P. Caudell a David Mizell vydali dokument, ve kterém popisují rozšířenou realitu jako systém, který sloužil pro zredukování chyb při vyrábění křídel letadel Boeing. [4,5]

1.3 Princip základní funkce rozšířené reality

Rozšířená reality vždy vykonává základní tři funkce, a to Sledování pozice, Rendering a následná Vizualizace na koncovém zařízení. Tyto funkce jsou propojené mezi sebou a všechny jsou spuštěny naráz v reálném čase. [4]

1.3.1 Sledovaní pozice

Hlídání pozice je jedna z hlavních vlastností rozšířené reality, díky ní může zařízení určit polohu s ohledem na okolní prostředí. Tato technologie umožňuje zařízení zjistit všech šest aktuálních úhlů pohledu uživatele, které se v reálném čase neustále mění a aktualizují při pohybu zařízení. Pozice je definovaná třemi souřadnicemi (x,y,z) a třemi úhly a to náklon, vychýlení a převrácení. Hlídání pozice využívá softwarové i hardwarové komponenty tak, aby výsledné hodnoty co nejvíce odpovídaly reálným hodnotám. [4] Jeden z nejkritičtějších vlastností je Pracovní místo, které definuje velikost oblasti, ve které systém funguje. Další vlastnosti hlídací pozice jsou vzorkovací frekvence, rozlišení a odezva

1.3.2 Rendering

Poté, co jsou zjištěny všechny souřadnice a úhly, je možné přidat a zvětšit obsah, který chceme do prostoru přidat. Dosazením těchto tří souřadnic a úhlů do kamery zobrazovacího zařízení dosáhneme cíle. Rendering je spravován pomocí softwarových knihoven a SDK, které jsou určeny specificky pro rozšířenou realitu. [4]

1.3.3 Vizualizace

Jedná se o poslední důležitou funkci, která cílí na dosažení konečného zobrazení iluze virtuálního světa na koncovém zařízení. Existuje více způsobů, jak tohohle dosáhnout. Tyto způsoby se dělí na dva hlavní proudy, a to na imerzivní nebo neimerzivní modalitu. Rozdíl mezi nimi je takový, že imerzivní modalita povinně vyžaduje fyzické nasazení headsetu. U neimerzivní modality se výsledný obraz zobrazuje na monitorech nebo na projektorech. [4]

1.4 Standardy rozšířené reality

Pro vývoj aplikací s rozšířenou realitou se využívají sady vývojových nástrojů. Tyto sady jsou vyvinuty na specifické platformy a zařízení a mají také rozdílné vlastnosti, výhody i nevýhody.

1.4.1 ARCore

Jedná se o platformu, která je exklusivně vytvořena pro operační systém Android od verze Android API level 24 (Android 7.0) a výše. Umožňuje mobilnímu zařízení interagovat s okolním prostředím a následně tyto informace zpracovávat. ARCore jako hlavní úkol zpracovává dvě činosti a to zejména sledování aktuální pozice mobilního zařízení s ohledem na pohyb za běhu a vytváření přídavného obsahu na obrazovce. K těmto účelům slouží ARCore motion tracking, která se používá pro kameru mobilního tařízení, aby detekovala body zájmu a sledovala jejich pohyb v reálném čase. Na vývoj aplikací s touto platformou je možné použít Unreal Engine, Unity nebo Android Studio. [6]

1.4.2 ARKit

Jedná se o platformu, která je exklusivně vytvořena pro operační systém od iOS 11 a vyšších verzí tohoto operačního systému. ARKit obsahuje v základu tři vrstvy, které jsou navzájem propojeny. První z ní je sledování, která má za hlavní úkol sledovat aktuální polohu zařízení, a to v reálném čase. Druhou vrstvou je rozklíčování scény, která má za úkol rozpoznat okolní prostředí, které snímá kamera mobilního zařízení. Pomocí těchto posbíraných dat se detekují klíčové body jako povrch země. Následně se do scény vloží požadovaný virtuální objekt. Poslední vrstvou je vykreslování samotných modelů na obrazovce uživatele. Na vykreslování je možně využít API Metal. K vytváření samotných aplikací se používají programy třetích stran jako například Unreal Engine nebo Unity. [7]

1.4.3 OpenXR

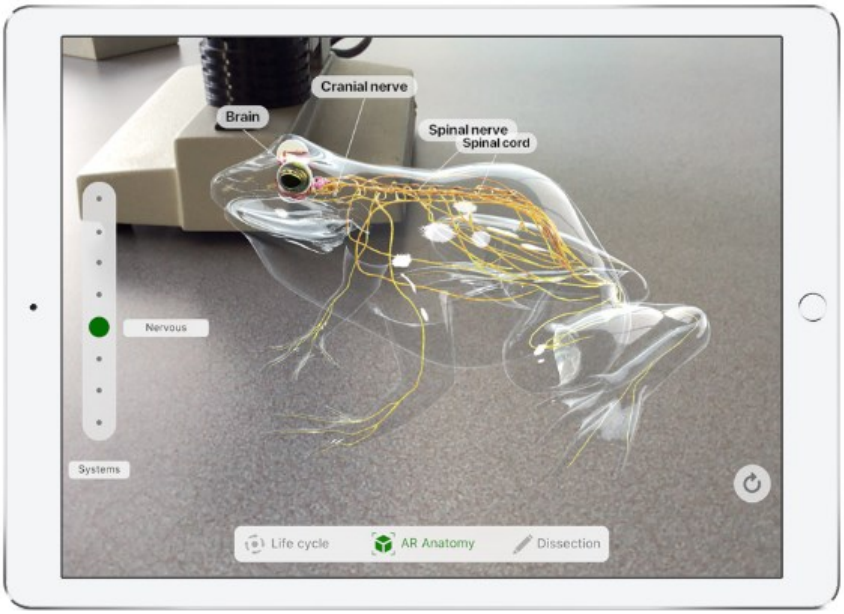
Jedná se o API, která je vydána s otevřeným zdrojovým kódem, aplikace lze exportovat na platformy Windows Mixed Reality, HoloLens 2 a Valve SteamVR.Toto API se zaměřuje na holografická zařízení. Hlavní výhodou této API je jednotný vývoj aplikace na všechny podporované platformy. [8] Na vývoj aplikací s touto API je možné použít herní engine Unreal Engine, Unity, Godot Engine 3, nebo vývojář může využít nástroje pro nativní vývoj aplikace. Mezi nejznámější aplikace využívající tohohle rozhraní jsou multiplatformní tituly Microsoftu, a to hra Minecraft a letecký simulátor Microsoft Flight Simulator.

1.5 Využití rozšířené reality v praxi

1.5.1 Využití ve vzdělávání

Jednou z významných oblastí, kde rozšířená realita využila svůj význam, je v oblasti vzdělávání. Studijní materiály pro školy mohou být umožněny prostřednictvím interaktivních 3D modelů s virtuálnímu objekty, což zlepšuje celkovou míru pochopení daných studijních materiálů a v některých případech zlepšuje zapamatování si daných objektů, které se v rozšířené realitě zobrazí.

Jedním z příkladů využití rozšířené reality je využití aplikace na hodiny dějepisu "Civilisation AR" od veřejnoprávní televize BBC. Tato aplikace umožňuje studentům poznávat a zkoumat historické a kulturní předměty přímo ve třídě a poskytuje jedinečnou příležitost prozkoumat předměty, jako je například Rossetská deska, egyptské sarkofágy, a další artefakty z mnoha různých historických období. Studenti mohou tyto artefakty uvidět na jejich hodině dějepisu v jejich původní velikosti, dozvědět se o nich základní informace a zacházet s nimi způsobem, který by byl v dřívějších dobách nemožný. [9]



Obrázek 1. AR Anatomy

1.5.2 Využití ve zdravotnictví

Rozšířená realita také postupně prohlubuje svoje využití v oblasti zdravotnictví s impozantními možnostmi transformace diagnostiky, léčby a interakce s pacienty. Jedním z

klíčových hráčů v této oblasti je produkt Microsoftu HoloLens, jsou to chytré brýle se smíšenou realitou, které se dají využít v lékařských postupech. [10] Tato technologie představuje inovativní přístup, kde se střetávají virtuální a reálný svět, poskytující lékařům a pacientům nový způsob provádění lékařských operací s informacemi, které dříve nebyly dostupné.

1.5.3 Surgical Navigation (Chirurgická Navigace):

Jednou z hlavních výstavních vychytávek využití produktu HoloLens ve zdravotnictví je takzvaná chirurgická navigace. Studie ukazují, že AR/MR systémy, zejména HoloLens 2, zlepšují přesnost, bezpečnost a účinnost chirurgických zákroků. Lékaři a operátoři mohou v reálném čase ve svých brýlích HoloLens zobrazovat medicínská data a zcela přesně napozicovat mechanické prvky s podporou virtuálních hologramů. To poskytuje lépe cílenou podporu pro hledání krevních cév a umožňuje lepší orientaci lékaře v operačním sále. [10]

Obrázek 2. Surgical Navigation

2 AR KÓDY

AR kód je dvoudimenzionální obrázek, který je navržený tak, že po naskenování čtečkou je na monitoru uživatel uměle vytvořen 3D objekt, který s daným AR kódem souvisí. Tyto kódy v praxi mají využití například při oblečení, kdy se uživatel při výběru bot naskenuje na svém mobilním zařízení AR kód, díky kterému se na zařízení ukáže na obrazovce 3D imitace těchto bot. [11]

2.1 Typy QR Kódů

2.1.1 Klasický QR Kód

Jedná se o dvoudimenzionální kód, který je určený k rychlému čtení, jak již napovídá anglický název Quick Response. Patentová práva vlastní společnost DENSO WAWE, která v minulosti uvolnila veřejnosti základní specifikace QR kódu, aby se stal co nejvíce využívaným širokou veřejností. Základní nevýhodou barokódů je nutnost čtení daného kódu pouze v jednom předurčeném směru, proto se japonský inovátor Mashiro Hara snažil přijít na řešení kódu, který by bylo možné číst z jakékoliv strany a pokud možno aby byl co nejvíce rychlý k přečtení čtecím zařízením. Tento problém se podařilo vyřešit tím, že přidal do vzoru černobílých čtverečků tři poziční značky na třech okrajích kódu, díky kterým lze kód číst ze všech stan a také využitím nejméně používaného poměru 1:1:3:1:1 šířek bílých a černých oblastí QR kódu.[12]

2.1.2 Princip čtení QR kódu

Když čtecí zařízení rozpozná tři poziční značky QR kódu, které jej ohraničují, tak se čtecí zařízení nejprve zaměřuje na spodní pravou část, která obsahuje modul identifikující verzi, ten obsahuje čtyř bitovou tabulku, pomocí které se indikuje, jakým ze čtyř možných způsobů je kód zakódován. Čtečka zařízení poté začne soustředit pozornost na modul indikující formát dalších osmi modulů, které předurčují velikost celého QR kódu. [13] Tímto způsobem se pokračuje do té doby, dokud se čtečka nedostane na Konečný Indikátor, který označuje konec standardních datových modulů. Poté se čtečka zaměří na moduly korekce chyb, které obsahují záložní data pro případ, kdyby část datových modulů byla poškozená.

Obrázek 3. Čtení QR kódu

2.1.3 SQRC

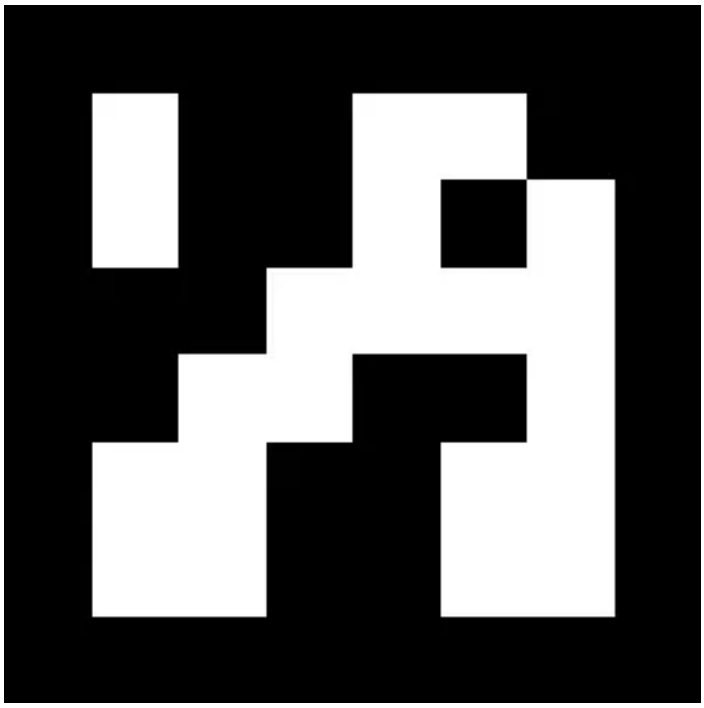
Jedná se o běžný QR kód, který umožnuje skrýt privátní klíč se soukromými daty pomocí určitého kryptografického klíče, který mohou odemknout pouze určité SQRC kompatibilní skenovací zařízení s náležitým privátním klíčem. Není zde potřeba šifrování a dešifrování. [14] Tento kód se používá například při distribuci lístků na koncerty, na velké akce s omezeným přístupem, u zásilkových služeb, a to kvůli prevenci podvádění nebo pozměňování QR kódu nežádoucím osobami. Nevýhodou je zde nutnost nákupu skenovacích zařízení.

2.1.4 FrameQR

FrameQR je běžný QR kód, který má nadstavbu obsahující oblast, která dovoluje tvůrci přidat do této vrstvy například obrázky, loga nebo jakoukoliv grafickou vizualizaci. Tvar oblasti, kde se tento obsah dá zobrazit je vymezen předem připravenými šablonami, které mají určitý formát, velikost, pozici a úhel. Cílem tohoto vylepšení je zaujmout větší pozornost kohokoliv, kdo se na daný QR kód podívá. Tento typ QR kódu se používá například u slevových akcí obchodů, upoutávek na koncerty nebo jakékoliv události a pro vizitky firem. [15] Nevýhodou tohoto QR kódu je snížená čitelnost pro skenovací zařízení a vyšší pravděpodobnost nenačtení kódu čtečkou. Jednoznačnou výhodou je, že vizuální vzhled přitahuje větší pozornost potenciálních uživatelů.

2.1.5 ArU-Code

ArU-Code je jedním z dvoudimenzionálních kódů, je sestaven z binární matice obsahující černá a bílá pole, která jsou ohraničena černým okrajem. Jedním z hlavách rozdílů oproti QR kódům je možnost naskenovat čtečkou více Aru-Code najednou v reálném čase. Další výhoda spočívá v tom, že tento kód je velmi rychlý z hlediska čtení čtecím zařízením a může být skenován z velké vzdálenosti. [16] V neposlední řadě se tenhle kód vyznačuje tím, že dokáže být rozpoznán čtecím zařízením i v slabě osvětlených místnostech. Tento kód se využívá ve velkých skladech díky možnosti vyhledávání balíku bez nutnosti skenovat každý balíček zvlášť.



Obrázek 4. ArU-Code

3 UNREAL ENGINE

Unreal Engine je herní engine vytvořený společností Epic Games, se stal nepostradatelným nástrojem pro vývoj her a virtuálních prostředí. Tento sofistikovaný herní engine se v průběhu let stal symbolem inovací a technologického pokroku ve světě herního průmyslu. Unreal Engine byl představen v roce 1998 jako součást hry "Unreal". Od té doby prošel řadou významných aktualizací a změn, přičemž každá nová verze přinesla s sebou revoluční vylepšení v oblasti grafiky a herního designu. Tento engine je vytvořen ve programovacím jazyce C++ a využívá ho jako výchozí možnost. Engine umožňuje vyexportovat hry na různé platformy, a to například Xbox Series X/S, Playstation 5, Windows, Mac OS X nebo Linux. [17,18]

Unreal Engine je program s otevřeným zdrojem kódu za určitých podmínek, které stanovuje společnost Epic Games. Mezi tyto podmínky patří povinnost hradit určitý podíl výdělku hry. Když hra bude vydělávat více než 3000$ za čtvrtletí, potom vývojář musí platit 5 % všech vydělaných peněz, co vývojář ze hry vydělal. [19]

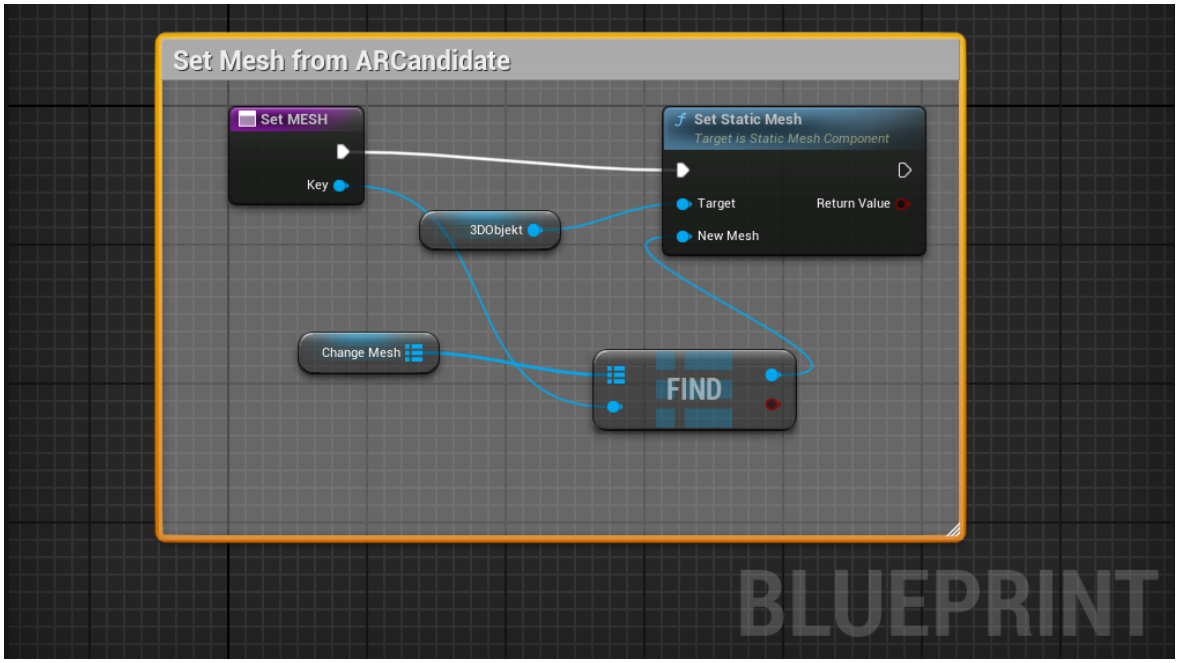
Unreal Engine je znám svými velmi náročnými požadavky na výkon grafické karty a procesoru daného zařízení, na které bude aplikace exportována. To naopak umožnuje velmi detailní grafické efekty a strhující animace. Jednou z klíčových charakteristik Unreal Engine je jeho grafický systém, který obsahuje pokročilý real-time rendering, stíny, osvětlení a postprocessing efekty. [20]

Důležitým prvkem je bezpochyby vizuálně skriptovací systém Blueprint, který umožňuje vývojářům vytvářet složitou herní logiku bez nutnosti znalosti programovacích jazyků, a to obzvlášť jazyka C++, který se používá na klasické programování. Tato přístupnost dokázala přilákat rozsáhlou škálu tvůrců od úplných začátečníků až po zkušené profesionály. [21]

3.1 Tvorba aplikace v Unreal Engine pomocí blueprintů

Modulární systém Blueprint v Unreal Engine je vizuální skriptovací jazyk pro tvůrce her. Tento inovativní nástroj poskytuje tvůrcům, level designerům a experimentátorům možnost vytvářet složitou herní logiku a interakce pomocí zjednodušeného vizuálního programování pomocí takzvaných nodes, které potom vývojáři navzájem propojují. Blueprinty jednoznačně zrychlují vývoj her a zjednodušují ho tak, že kdokoliv bez znalostí jediného programovacího jazyka je schopen vytvořit složitou hru, aniž by napsal jediný řádek kódu.

Blueprinty ve Unreal Engine se rozdělují do čtyřech základních typů, které se navzájem odlišují a vždy slouží k rozdílným účelům. [22].



Obrázek 5. Ukázka Blueprintu

3.1.1 Level Blueprint

Prvním z nich je Level Blueprint, ten je automaticky vytvořen ke každému levelu, slouží k tvorbě průběhu dané úrovně pomocí sekvencí, které se nazývají Function Calls můžeme vytvářet rozdílné funkce. K tvorbě animací se zde používají nástroje Maitinee. Pro kontrolování průběhu programu zde slouží Flow Control operations. [22]

3.1.2 Blueprint class

Tento Blueprint se používá k přidávání funkcí k samotnému ději hry. Vývojář aplikace zde dodává funkce, které pomocí Nodes propojuje mezi sebou. Mezi Blueprint class patří pět běžné rodičovských tříd. Tyto rodičovské třídy se nazývají Actor, Pawn, Character, Game Mode a PlayerController. [22]

3.2 Porovnání s ostatními herními enginy z hlediska Rozšířené reality

3.2.1 Unity

K tvorbě AR aplikací lze použít herní engine Unity 3D. V základní verzi podporuje Apple ARKit na platformu iOS a Google ARCore na operačním systému Android. Hlavním

rozdílem oproti Unreal Engine je zabudovaný nástroj Unity Mars, který zjednodušuje tvorbu vývojáři AR aplikací. Tento nástroj nabízí simulační mód, kde může vývojář používat předpřipravené prostředí a pomoci nich simulovat reálný běh aplikace. Důležitou součástí tohoto balíku je funkce Plane-language authoring, pomocí které může vývojář přednastavit chování aplikace v určitých situacích, například AR naskenuje dveře a aplikace je automaticky otevře.[23]

3.2.2 Godot Engine

Dalším z konkurenčních herních enginů je Godot Engine s otevřeným zdrojovým kódem pod licencí MIT. Tento engine oficiálně podporuje otevřený formát OpenXR, OpenVR SDK, Oculus SDK, OpenHMD. Tyto formáty se dají stáhnout z oficiálního repozitáře pluginů Godot Engine s názvem GodotVR Repository Ostatní standardy jako ARCore a ARKit jsou neoficiálně podporovány pomocí uživatelských pluginů. V Godot Engine 3.0 byl implementován AR/VR server, ten využívá programovací jazyk GDNative. Vyzžívá se zde čtyřech nových nodes, které byly vytvořeny pro tvorbu aplikací s rozšířenou realitou. Tyto nodes obsahují pozici, kde se daný uživatel nachází, [24]

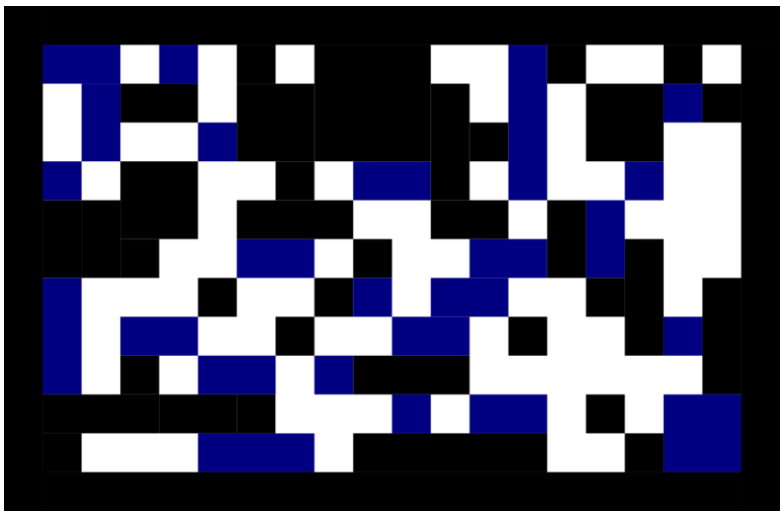
II. PRAKTICKÁ ČÁST

4 TVORBA VLASTNÍHO AR KODU

Pro účely této práce byly vytvořeny dva vlastní AR kódy, každý z nich má jiné zaměření. Tyto kódy splňují několik základních požadavků. Mezi ně patří rychlost načtení AR kódu čtecím zařízením, kde byla testována a porovnána v závěrečné testovací kapitole mezi všemi AR kódy. Dalším důležitým požadavkem je samotná kapacita dat, které určuje celkový maximální počet alfanumerických znaků, čísel, bitů nebo jiného daného typu dat. Dalším parametrem je čitelnost kódu, který je vždy čitelný čtecím zařízením za běžných světelných podmínek. Během tvorby těchto kódů byly využity běžně dostupné nástroje s otevřeným zdrojovým kódem, jako je například vektorový program Inkscape nebo rastrový program GIMP. Pro tvorbu 3D modelů, které se zobrazují po přečtení čtečkou AR kódů na obrazovce uživatele byl využit modelovací program Blender. Následně tyto kódy na byly porovnány s klasickým QR kódem, a to hlavně z hlediska rychlosti čtení kódu.

4.1.1 Tritový kód

Hlavním zaměřením tohoto kódu je co největší rychlost načtení čtečkou kódu a zároveň co největší datová hustota. Z tohoto důvodu zde byly přidány takzvané trity z trojkové soustavy, které mohou nabývat třech různých hodnot místo dvou u binárních čísel ze soustavy dvojkové. První kód, který byl navržen obsahuje černý okraj s tloušťkou jednoho tritu, který má v sobě datové pole v rozsahu 198 tritů (313,83 bitů) v rozložení jedenáct na výšku a osmnáct na šířku. Trity mohou být znázorněny černou, modrou nebo bílou barvu. Trity uvnitř černého okraje jsou různě rozmisťovány pomocí generátoru pseudo náhodných čísel.



Obrázek 6. Tritový kód

4.1.2 Barevný obrázkový kód

Hlavním zaměřením tohoto kódu je jeho unikátní vzhled a snadná rozpoznatelnost. To znamená, že pozadí tohoto kódu bylo složeno z jakéhokoliv barevného obrázku. V tomto konkrétním případě kód má vzhled imitující 3D model lišky, který byl vytvořený v modelovacím programu Blender, který se po přečtení AR kódu zobrazí na obrazovce uživatele. Na pravém horním a levém dolním okraji bude nápis AR označující kód, aby ho bylo možné snadněji identifikovat ze všech stran.

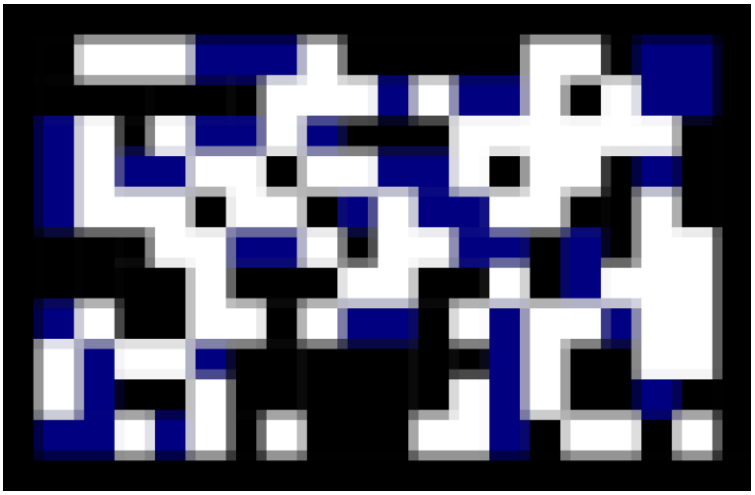
Obrázek 7. Barevný obrázkový kód

4.2 Testovací kódy

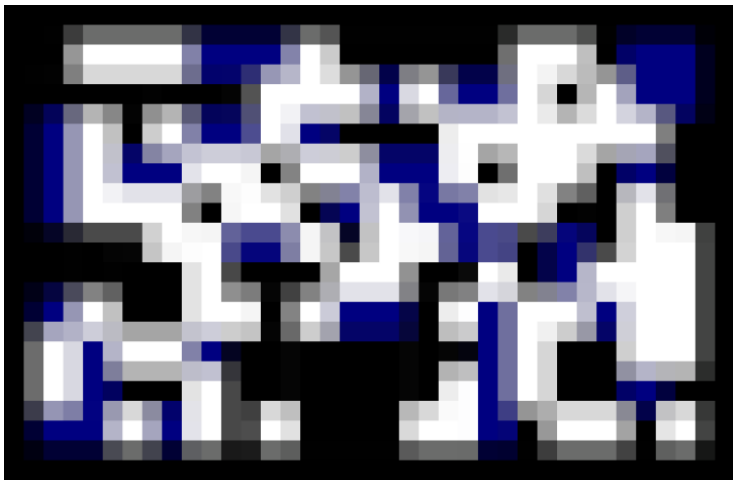
Aby bylo možno porovnat kódy mezi sebou z hlediska rychlosti načtení čtecím zařízením, byly vytvořeny kódy, u kterých byl pixelizován obrázek z toho důvodu, aby se zjistili vliv na rychlost načtení AR kódu. Pro tento účel byl využit rastrový editor GIMP 2.10.34.

4.2.1 Tvorba testovacích kódů v programu GIMP

Původní obrázek Tritového kódu byl otevřen v programu GIMP. V hlavním menu Filtry/Rozmazání byla vybrána položku Kostičkovat. Zde byla nastavena šířka i výška bloků na hodnotu 16. Tato volba byla potvrzena tlačítkem OK. Poté v hlavním menu bylo kliknuto na Soubor/Export as a poté byl vyexportován tento obrázek. Pomocí klávesové zkratky Ctrl + Z byl navrácen původní obrázek a následně zopakovány předchozí kroky s tím rozdílem, že zde byla nastavena šířka a výška bloků na hodnotu 32. Tento obrázek byl vyexportován. Výsledné obrázky jsou níže.



Obrázek 8. Tritový kód s pixelizací 16 bloků

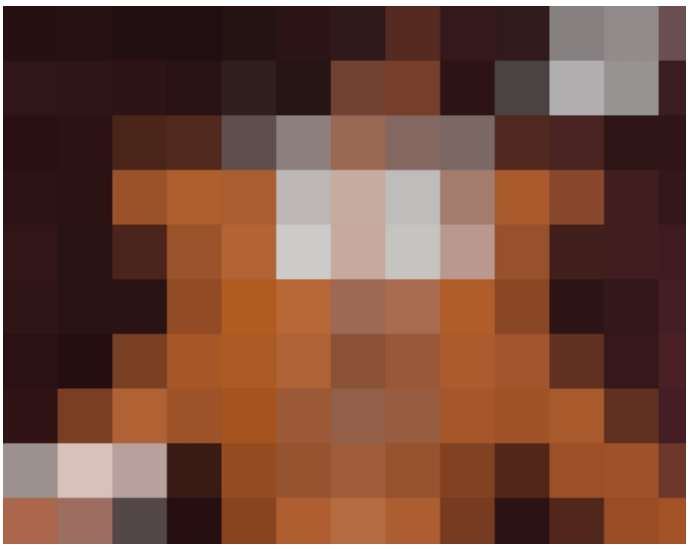


Obrázek 9. Tritový kód s pixelizací 32 bloků

V tomto případě můžeme vidět, že s postupnou pixelizací přibývají pixely obsahující datovou informaci, tedy jedná se o paradox.

Následně byly vytvořeny obdobné testovací kódy pro barevný obrázkový kód, postup byl stejný jako v předchozím případě. Velikost bloků byla nastavena na hodnotu 16 pro výšku i šířku. Výsledné obrázky následně byly uloženy do počítače.

Obrázek 10. Barevný obrázkový kód s 16 bloky



Obrázek 11. Barevný obrázkový kód s 32 bloky

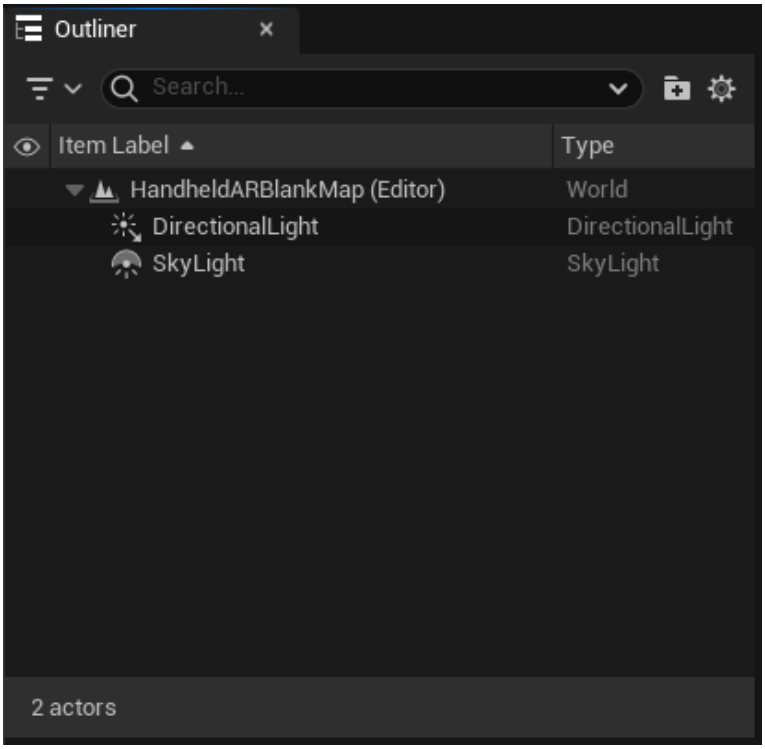
V případě tohoto testovacího kódu lze vidět, že při postupné pixelizaci klesá počet pixelů obsahující datovou informaci, tedy v tomhle případě by měla být rychlost načtení AR kódu s postupnou pixelizací rychlejší než u základního obrázkového kódu.

5 TVORBA TESTOVACÍ PROGRAMU

V této kapitole byla vytvořena implementace AR kódu. To znamená, že program bude muset umět zvládnout naskenovat daný AR kód, poté uživateli na jeho obrazovce přiřadit správný 3D model nastavený k danému AR kódu a následně jej zobrazit. Byl využít program Unreal Engine 5.3.2, obrázky AR kódu, který byly vytvořeny. Pro tvorbu 3D modelů byl využiit modelovací program Blender 3.6 LTS. K testování byl využíit mobil s operačním systémem Android, který měl oficiální podporu Google ARCore a měl nainstalovaný v sobě minimálně verzi Android 7.0. V tomto konkrétním případě byl využit mobil Motorola Moto G42 6 GB XT2233-2 s oficiální stock verzí operačního systému Android 13. V druhé části této kapitoly výslednou aplikaci byla exportována přímo do testovacího mobilu, kde bylo vyzkoušeno, zda program splnil všechny základní požadavky.

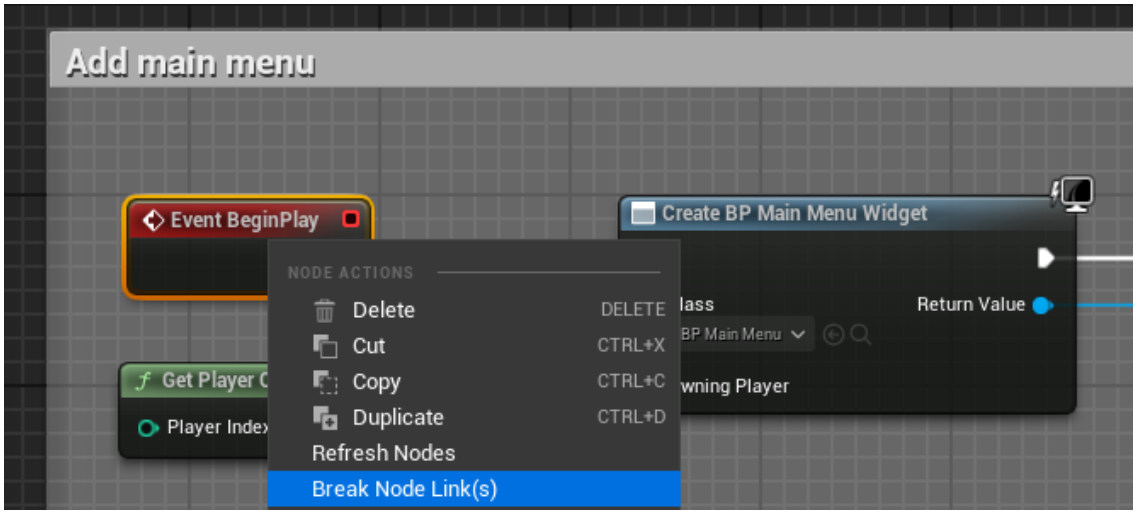
5.1.1 Vytvoření projektu

Byl otevřen program Unreal Engine 5.32 a vytvořen nový projekt s názvem bak2, byla vybrána zároveň šablona Handeheld AR, která se nacházela ve výchozích šablonách Games. Poté bylo zmáčknuto tlačítko Create, projekt byl teď vytvořen. Nyní se projekt sám otevřel, z nabídky Outliner v podadresáři HandeheldARBlankMap byly vymazány všechny zbytečné věci tak, že pouze zůstal aktér SkyLight a DirectionalLight.



Obrázek 12. Outliner List of Items

Byl otevřen ContentDrawer, kde se nacházely všechny složky projektu, byla rozkliknuta složku Blueprints a podadresář GameFramework, kde se nacházel vygenerovaný Blueprint soubor BP_ARPawn který byl posléze otevřen. Následně byl vyhledán Blueprint Add main menu a zde u položky Event BeginPlay bylo odebráno spojení pomocí volby Break Node Link(s), Blueprint byl uložen a zkompilován, poté bylo prostředí programu vráceno do Hlavního Editoru.

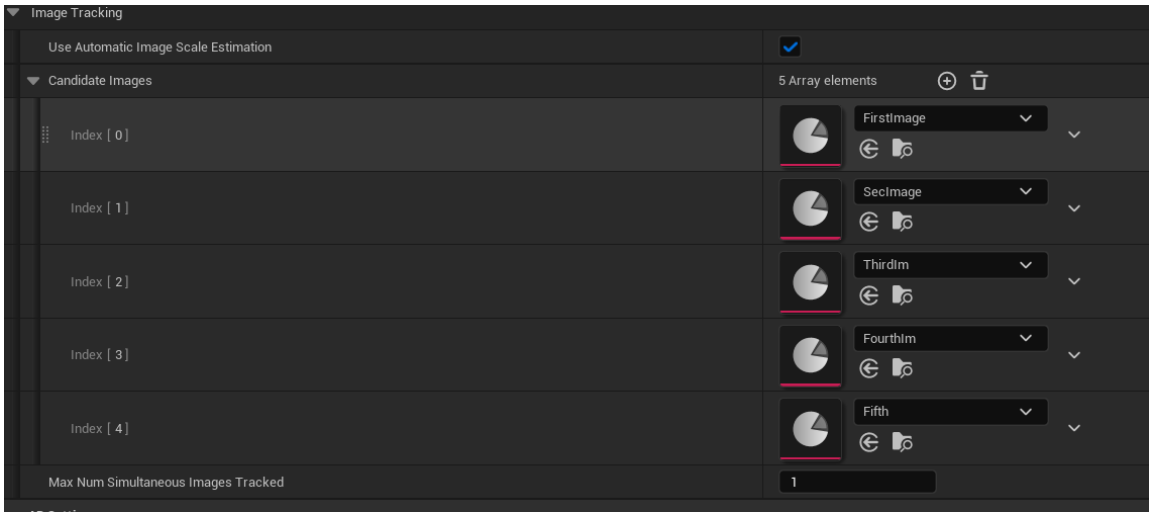


Obrázek 13. Blueprint ARPawn

5.1.2 Tvorba implementace AR kodu

Byl otevřem Content Drawer do složky projektu a vytvořen zde nový Data Asset z menu Miscellaneous, tento asset byl pojmenován jako FirstImage. Poté byly vloženy do projektu obrázky, který byly vytvořeny v předchozí kapitole pomocí jednoduchého přetažení obrázku z průzkumníku souborů do složky projektu. Poté byl otevřen asset FirstImage, zde byla vložena texturu AR kódu, kterou byla naimportována. Vyplněna zde také byla položka Friendly Name textem arcodeimage, protože tato položka musela být povinně vyplněná. Poté bylo vráceno prosředí programu do složky projektu v Content Drawer, zde byl otevřen Data Asset D_ARSessionConfigFirstImage, v položce Candidate Images ve oddílu Image Tracking byl vložen nový ellement, do kterého byl zvolen Data Asset First Image a následně daný asset byl uložen. Potom byly vytvořeny další položky Data Asset s například jménem SecImage, zde byly provedeny stejné kroky jako s předchozím Data Asset FirstImage s tím rozdílem, že zde byl přidán jiný obrázek AR kódu a změněna proměnná friendly name. V Data Asset D_ARSessionConfig do rámečku Candidate Images byly vloženy všechny Data Asset, kde byly uložené AR kódy. Také zde byla nastavena u položky Max Num

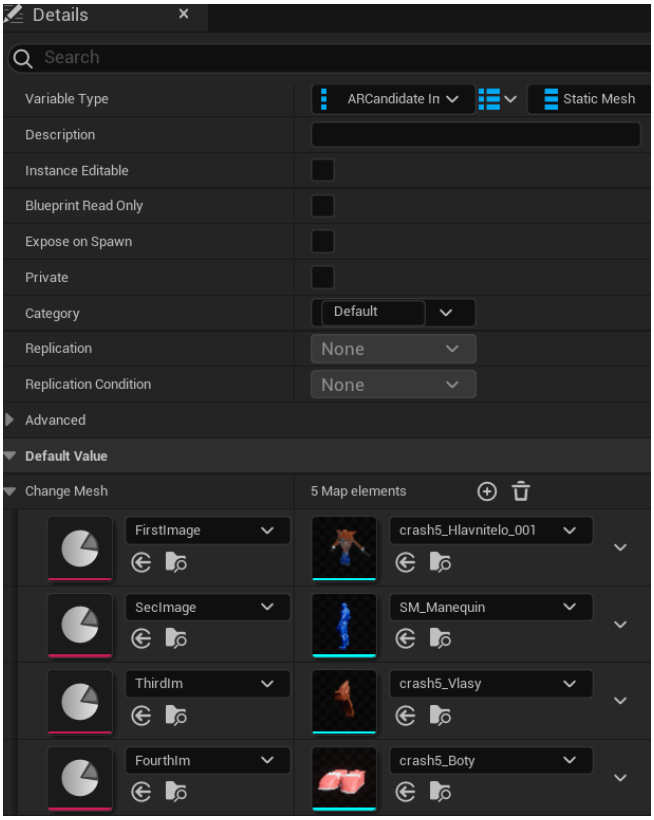
Simulated Images hodnota 3, kterou byla proměňována v závislosti na dané situaci programu.



Obrázek 14. Nastavení Data Asset D_ARSessionConfig

5.1.3 Tvorba funkce SetMESH

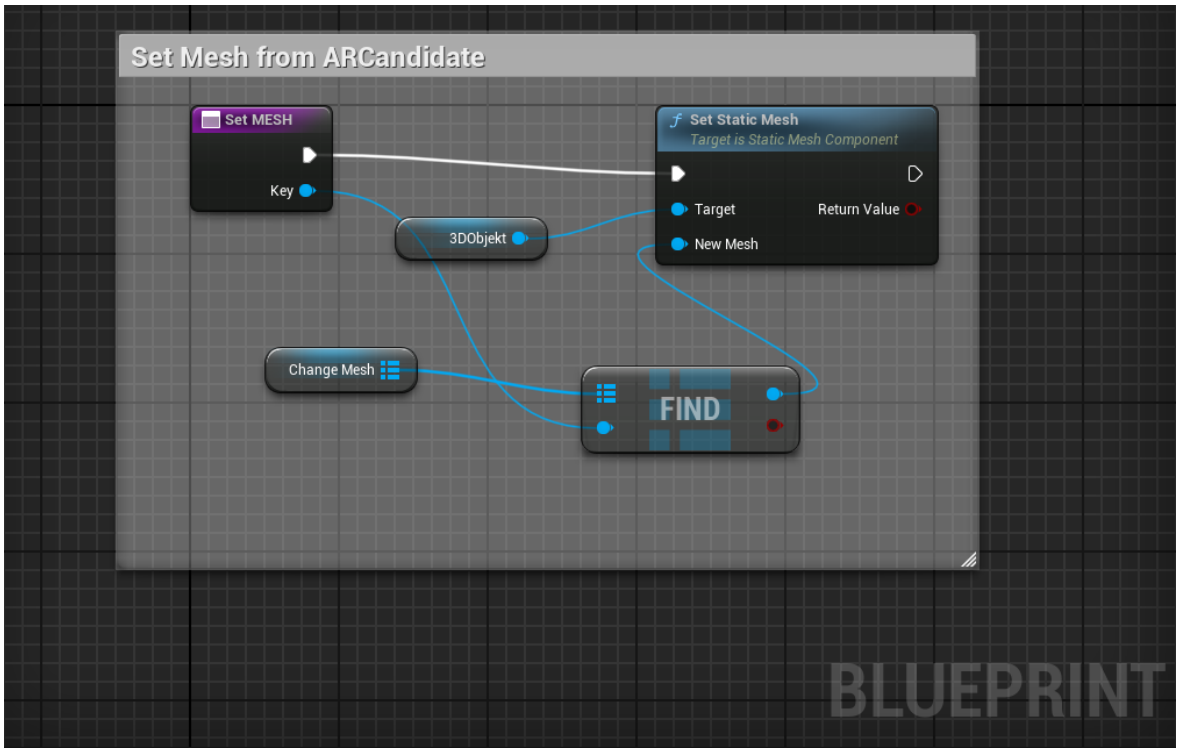
Byl otevřen ContentDrawer v root adresáři projektu, byla rozkliknuta složka Blueprints a poté podsložka GameFramework. Pomocí pravého stisknutí tlačítka myši byl vytvořen Blueprint Class, který byl pojmenován AR_Model, který posléze byl otevřen. V panelu MyBlueprint byla vytvořena nová funkce SetMESH, která sloužila k detekování klíče ARCandidate, pomocí kterého se můžou měnit Static Mesh podle přednastaveného AR kódu. Poté v panelu MyBlueprint byly vytvořena nová proměnná, ta byla pojmenována na ChangeMesh. Poté bylo kliknuto na vytvořenou proměnou a program zobrazil detaily proměnné, proměnná byla nastavena na ARCandidateImage Object Reference. Vpravo se nacházela položka Type of the variable, ta byla nastavena na Map. Číselný typ byl nastaven na Static Mesh Object Reference. V detailech proměné Change Mesh byly přidány defaultní odkazy na AR kódy a jejich přináležející Static Mesh modely.



Obrázek 15. Přiřazení 3D modelů k AR kódům

Následně byl otevřen Blueprint proměnné SetMESH, pomocí levého tlačítka byla přenesena proměnná ChangeMesh a cílový Static Mesh s názvem 3DObjekt do Blueprintu. Poté byl

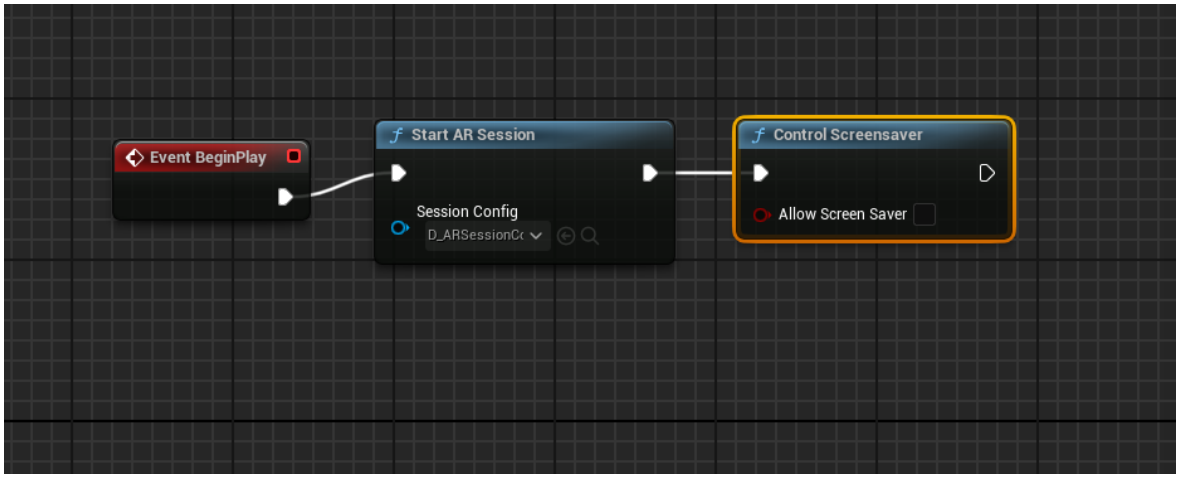
přidán další node s názvem Find a node Set Static Mesh. Tyto nodes byly propojeny následovně podle obrázku.



Obrázek 16. Funkce SetMESH

5.1.4 Level Blueprint

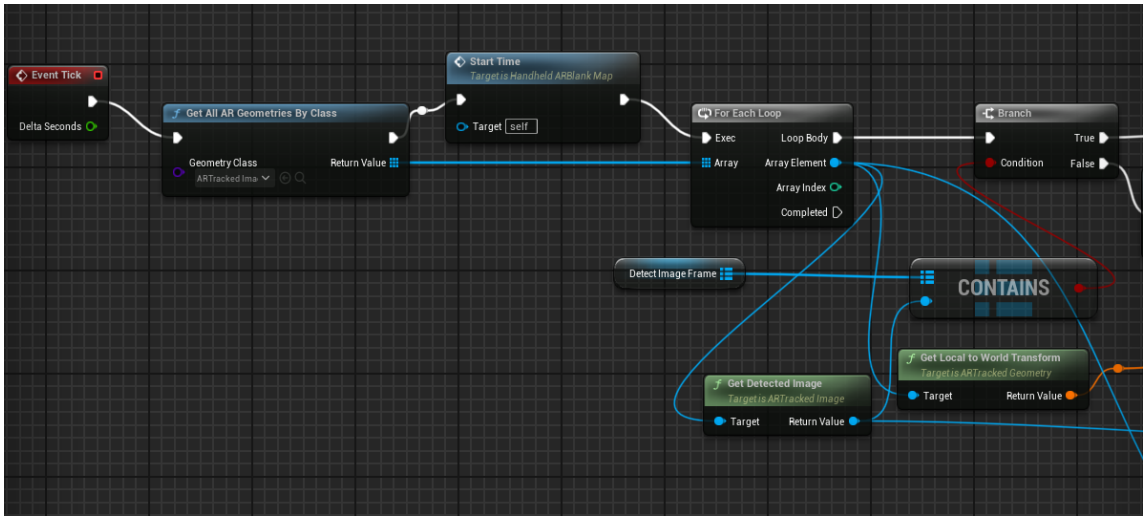
V horní nabídce programu byla otevřena v List of World Blueprints položka Open Level Blueprint, zde se zobrazil EventGraph, který byl upraven. Pomocí pravého tlačítka byl přidán node Start AR Session a Control Screansaver. U node Start AR Session byla nastavena Session Config na D_ARSessionConfig. Následně všechny nodes byly propojeny podle následujícího obrázku.



Obrázek 17. Event BeginPlay Blueprint

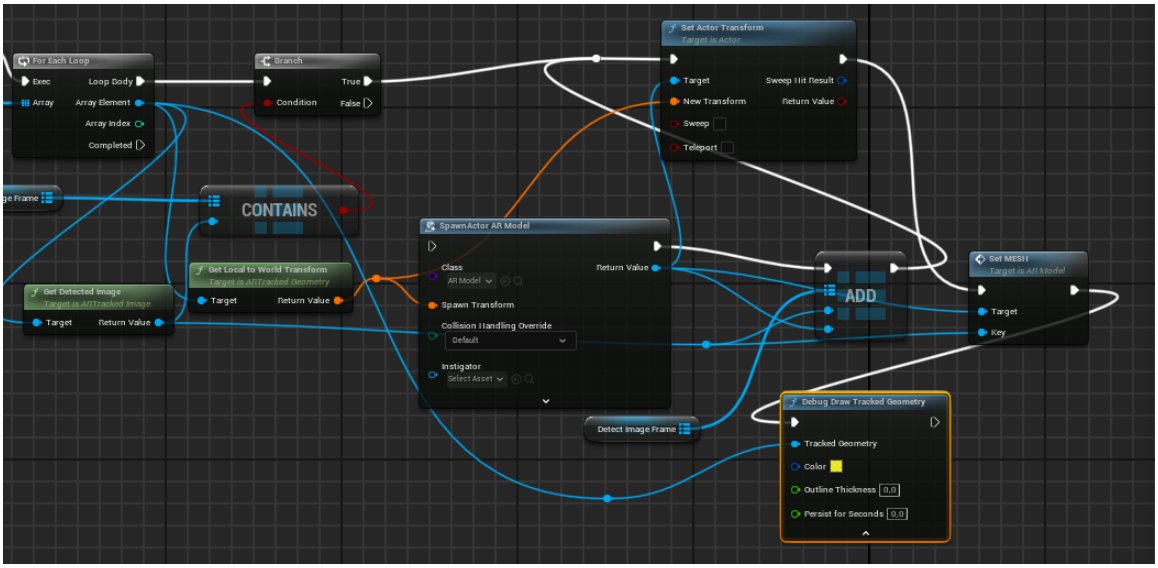
V panelu My Blueprints byla vytvořena nová proměnná, která byla nazvána jako DetectImageFrame, v detailech této proměnné byl nastaven pin type na ARCandidate Image Object Reference, value type byl nastaven na AR Model Object Reference.

Poté byl přidán do Blueprintu pomocí stisku pravého tlačítka myši nový Node s názvem Get All AR Geometries By Class, kde byl nastavem Geometry Class na ARTrackedImage. Pomocí levého tlačítka myši byla přidána proměnná DetectImageFrame do Blueprintu. Následně byly přidány další Node For Each Loop, Node Branch, Node Get Detected Image a Node Get Local to World Transform, které následně byly propojeny mezi sebou podle obrázku níže. Vytvořené spojení zajistilo, že se bude kontrolovat, zda se změnil AR kód na vstupu. Pokud se tedy změní AR kód, tak se podmínka přesune do větve false, kde posléze lze měnit 3D model, který se bude zobrazovat na koncovém zařízení.



Obrázek 18. Blueprint AR Tracking Image

Do Blueprintu byl následně přidán Node Spawn Actor from Class, položka Actor Class Reference byla nastavena na AR Model. Posléze byly přidány Nodes Set Actor Transform, funkce SetMesh, Debug Draw Traced Geometry, Add a pomocí levého tlačítka myši byla přetáhnua proměnná DetectImageFrame do Blueprintu. Ty následně byly propojeny mezi sebou podle obrázku níže. U Node Debug Draw Tracjed Geometry bylo možno nastavit libovolnou barvu, která se zobrazuje kolem skenovaného AR kódu. Proměnná Outline Thickness byla přenastavena z původní hodnoty 5 na hodnotu 0,2 z toho důvodu, aby nebyl výsledný okraj na AR kódu příliš tlustý.

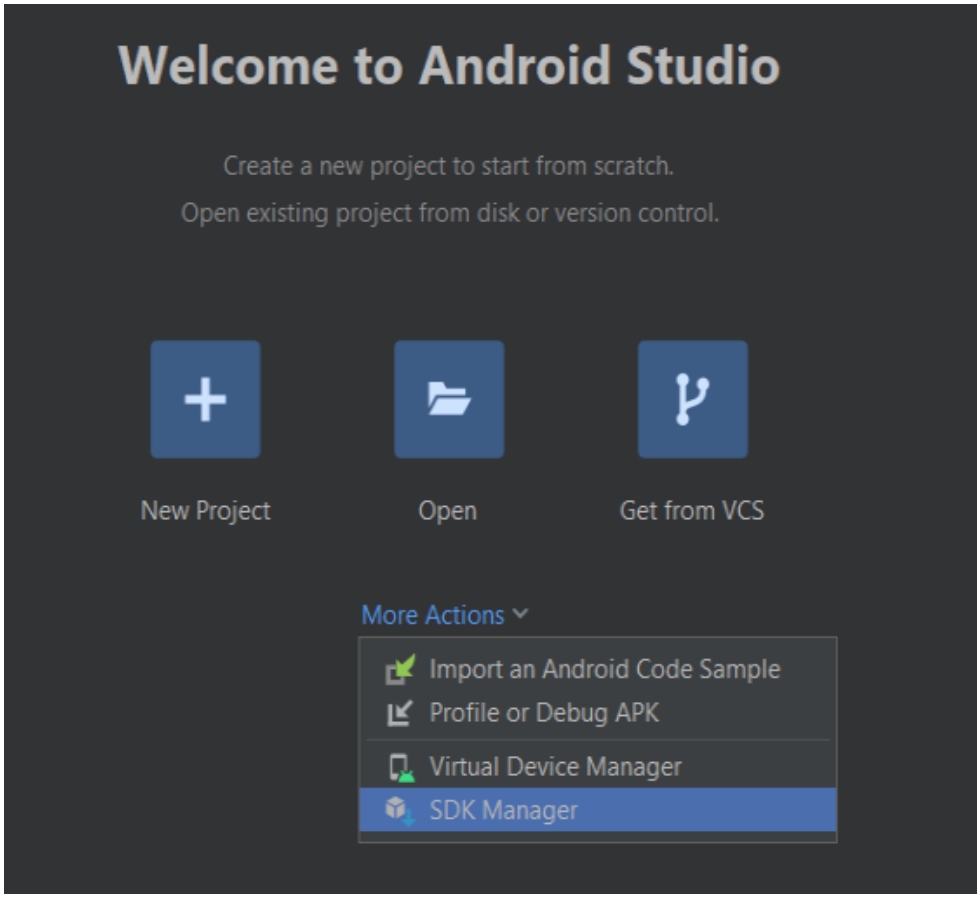


Obrázek 19. Konečný AR Tracking Image Blueprint

Blueprint byl uložen a zkompilován, celý projekt byl uložen. Poté bylo prostředí programu zpátky vráceno na úvodní projektovou obrazovku, zde byla v následující podkapitole aplikace vyexportována do mobilního zařízení.

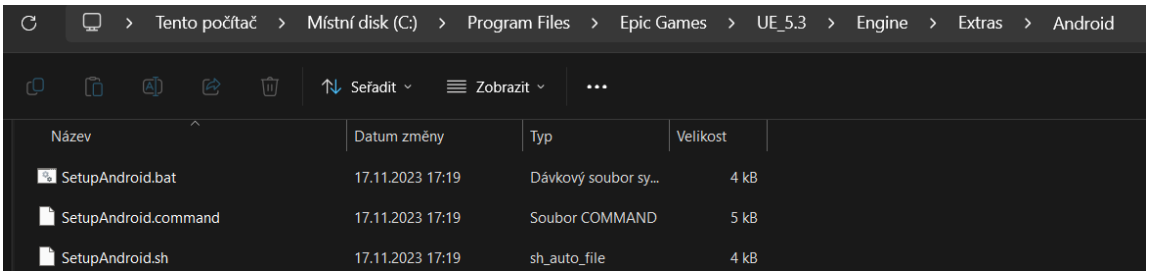
5.2 Nahrání programu do mobilního zařízení

Nejnovější verze Android Studio Hedgehog byla stažena, která je doporučená tvůrci Unreal Engine verze 5.3.2. Při otevření programu se zobrazil uvítací dialog, zde bylo rozkliknuto tlačítko More Actions a vybrána volbu SDK Manager. Poté zde bylo vybráno tlačítko Edit a nainstalována nejnovější verze Android SDK.



Obrázek 20. Přivítací obrazovka Android Studio

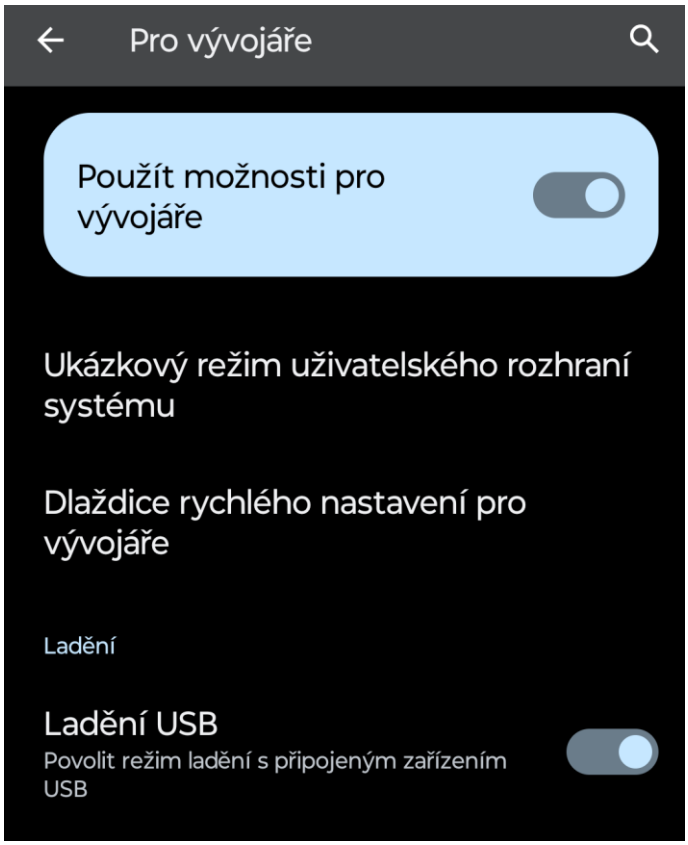
Poté, co bylo správně nainstalované Android Studio, tak byl otevřen root adresář Unreal Engine, zde bylo rozkliknuto Engine/Extras/Android a spušťen soubor SetupAndroid v závislosti na operačním systému, který při vývoji byl využit. V tomto konkrétním případě byl využíván operační systém Windows, proto byla spuštěna varianta SetupAndroid.bat a následně byl celý proces ponechán doběhnout do konce.



Obrázek 21. SetupAndroid ve File Explorer

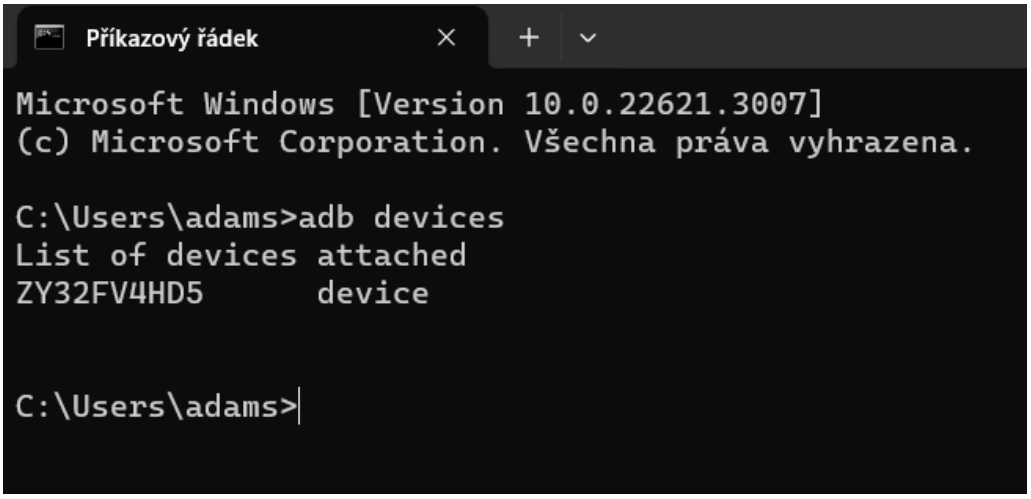
Poté bylo připraveno Android zařízení, na které byl program nahrán. Zařízení bylo vybráno s ohledem na podporu minimálně API level 26, ve kterém byly podporovány ARCore služby a Unreal Engine. Zařízení bylo odemknuto a v nastavení v položce informací o telefonu sedmkrát kliknuto na číslo nastavení. Po sedmi kliknutích se zobrazila zpráva, že se odemkly

nastavení vývojáře. Zde v nastavení vývojáře v horní nabídce byla povolena kolonka Ladění USB.



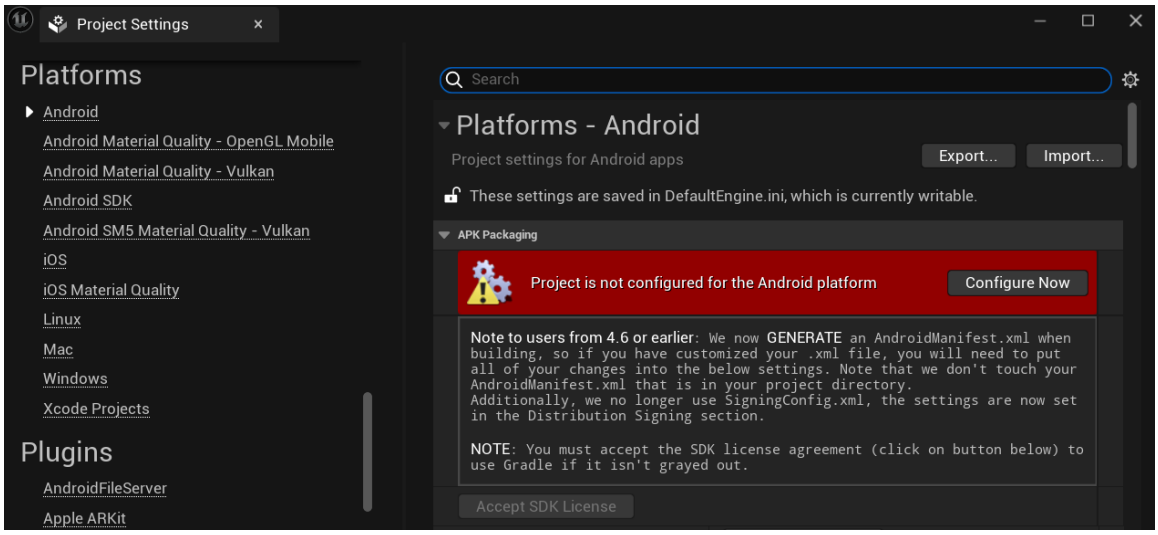
Obrázek 22. Nastavení ladění USB

Zařízení posléze bylo připojeno k počítači pomocí portu USB, kde byly předem správně nainstalovány USB ovladače zařízení a Android Debug Bridge (ADB). Následně byl otevřen příkazový řádek nebo terminál v závislosti na operačním systému, který má daný počítač nainstalovaný. Byl zde vložen příkaz adb devices, na zařízení se objevilo dialogové okno, kde byla povolena volba USB debugging. Zařízení bylo poté v ADB správně autorizované.



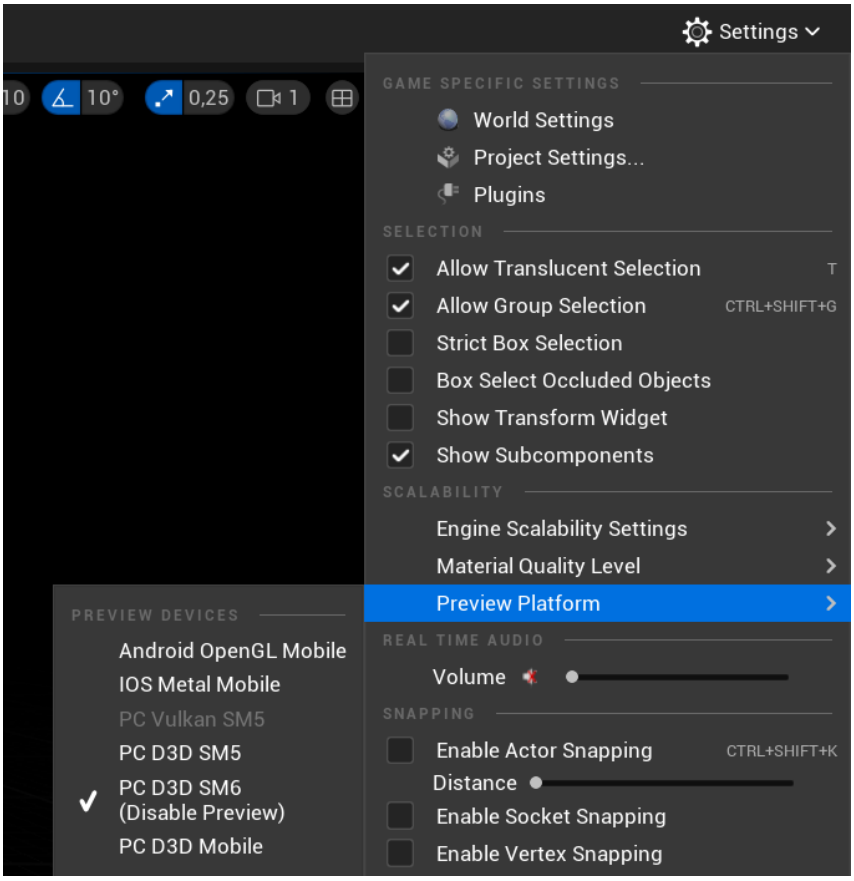
Obrázek 23. Autorizace zařízení ADB

V Unreal Engine byl poté otevřen projekt, který byl vytvořen v předchozí podkapitole. V nastavení projektu Platforms/Android bylo kliknuto na tlačítko Configure Now a poté potvrzena volba SDK License.



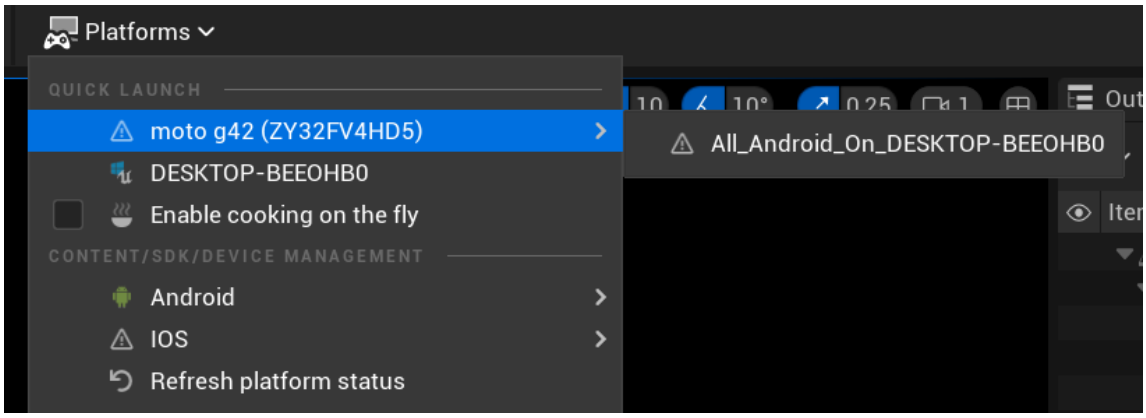
Obrázek 24. Potvrzení SDK License

Následně bylo zavřeno nastavení projektu a posléze v Setings/Previw Platform byla nastavena volbu Android OpenGl Mobile, protože API Vulkan v současné době nepodporuje rozhraní Google ARCore.



Obrázek 25. Volba Platformy

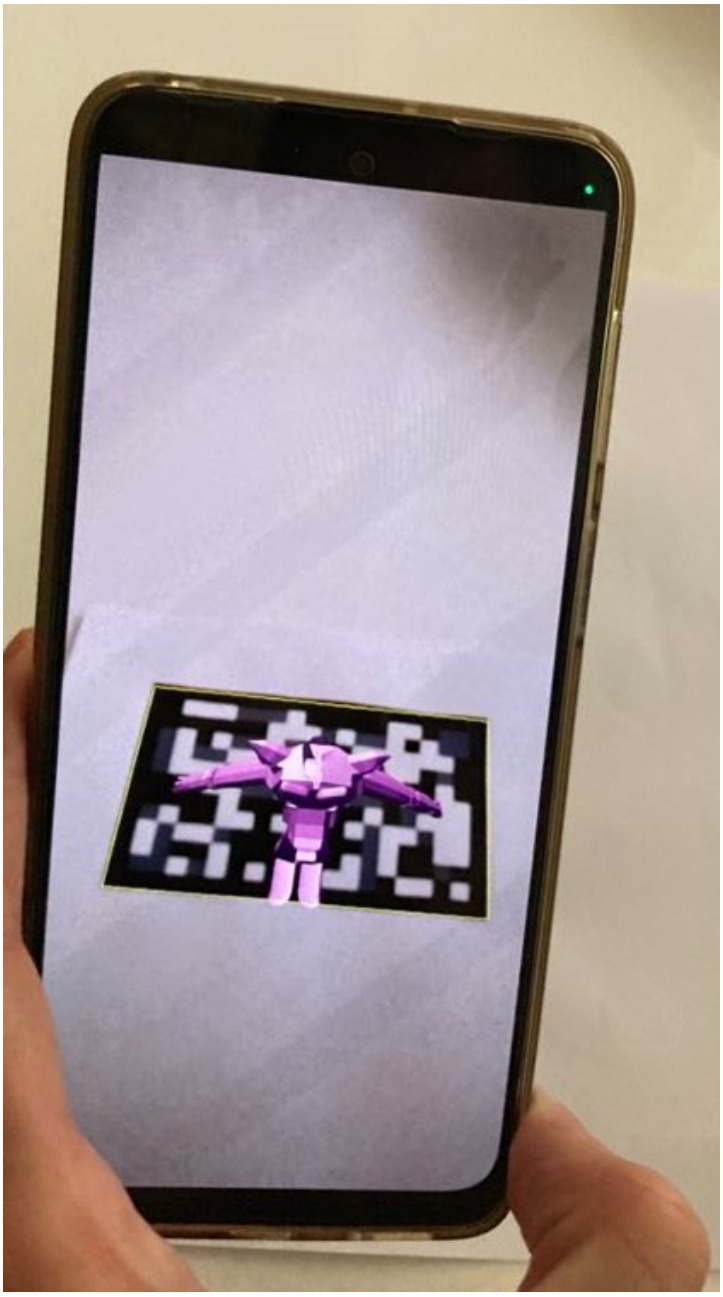
Následně byl označen level, který byl exportován na mobilní zařízení. V hlavní nabídce bylo kliknuta položka Platforms a vybráno zařízení, na které se program nahrál, v tomto konkrétním případě byla vybrána volba Android zařízení, pokud se zařízení nezobrazovalo v nabídce, byla zde vybrána volba Refresh platform status, aby Unreal Engine zařízení znovu zaregistroval a následně zobrazil.



Obrázek 26. Export aplikace na Android zařízení

Program byl úspěšně nahrán na zařízení, program se sám automaticky spustí a můžeme na něm zkoušet funkčnost všech AR Kódu a to tím, že zkusíme všechny kombinace

předehraných AR kódu a zkontrolujeme, jestli 3D modely sedí s daným AR kódem. Pro ukončení debuggování aplikace klikneme na tlačítko Cancel. Program se potom sám vypne a na zřízení se zobrazí domácí menu.



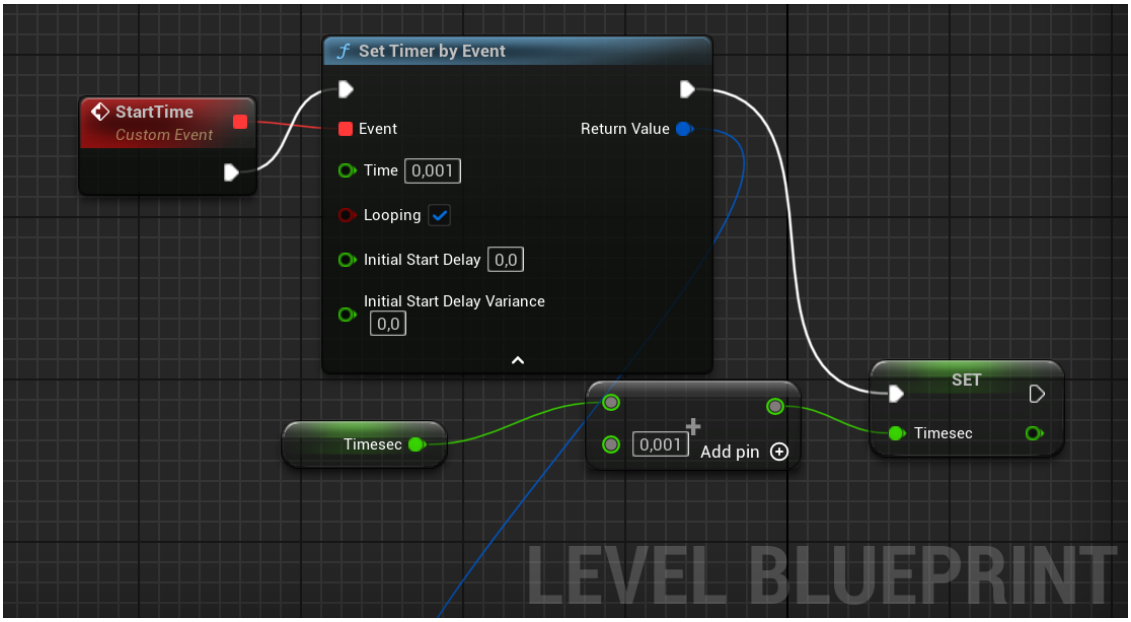
Obrázek 27. Finální funkční aplikace v provozu

6 TESTOVÁNÍ VYTVOŘENÉHO AR KÓDU

V této kapitole byla vytvořena testovací aplikace, která umožnuje změřit rychlost načtení různých AR kódů v sekundách a následně klasického QR kódu. Za účelem tohoto byly vytvořeny časovače, které měří v tisícinách sekundy z důvodu co nejvyšší přesnosti měření. Nakonec této kapitoly byly tyto kódy vzájemně porovnány podle těchto parametrů.

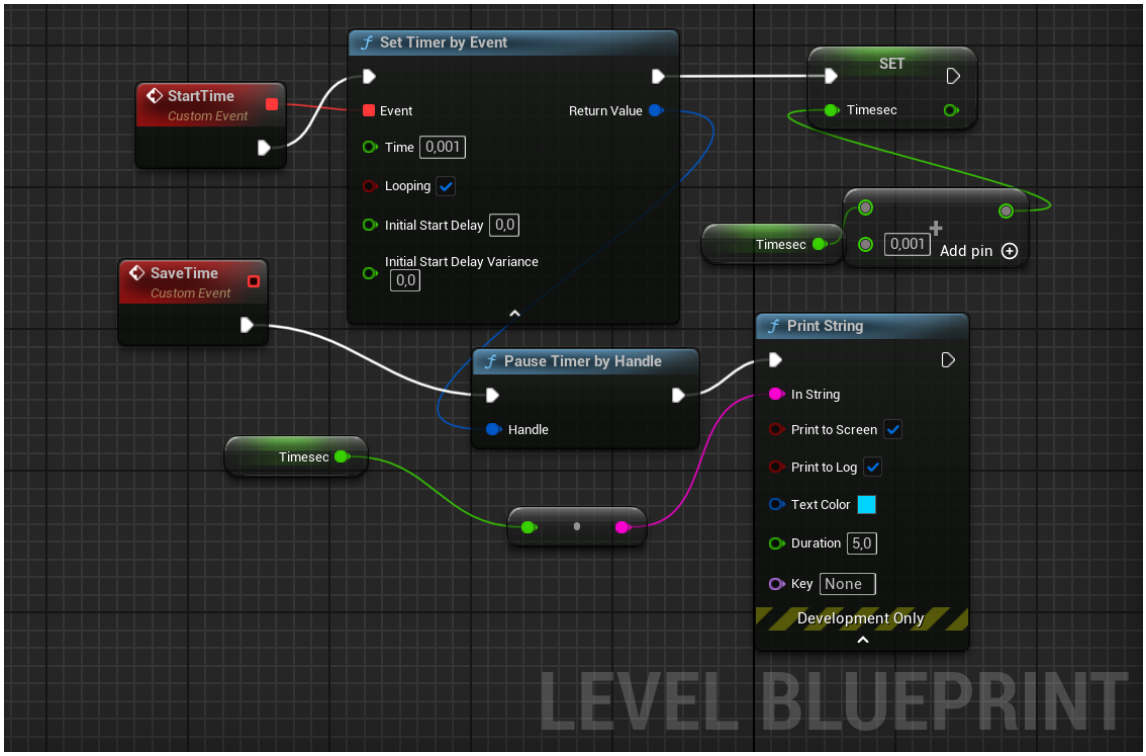
6.1 Blueprint časovače

Byl otevřen Level Blueprint vytvořeného projektu Handeheld AR. Zde byla vytvořena funkce, která měla za cíl spustit časovač při skenování AR obrázku, poté tento časovač přestane čítat čas při konečném zjištění AR kodu. Za tímto účelem byla vytvořena v panelu MyBlueprint nová proměnná s názvem Timesec, které byl přiřazen datový typ Float. Tato proměnná byla vytvořena k následnému čítání sekund. Poté byly vytvořeny v Blueprintu pomocí pravého tlačítka myši dva custom eventy, jeden slouží ke spuštění časovače v určitém bodě, tento event byl nazván StartTime. Druhý custom event s názvem SaveTime slouží k opačnému účelu, tedy k zastavení čítání sekund. Posléze byl přidán node Set Timer by Event, kde byla nastavena položku Time na hodnotu 0,001 z toho důvodu, aby bylo měření co nejvíce přesné. Poté ještě byla zakliknuta položka Looping na opakování procesu. Z panelu MyBlueprint byla přetáhnuta z pole variables proměnná Timesec pomocí jednoduchého přetáhnutí levého tlačítka myši. Poté byl přidán node Operators Add, kde byla nastavena hodnota na 0,001, aby se časovač čítal pomocí tisícin sekundy. Následně znovu do Blueprintu byla přidána proměnná Timesec. Nodes byly propojeny podle následujícího obrázku.



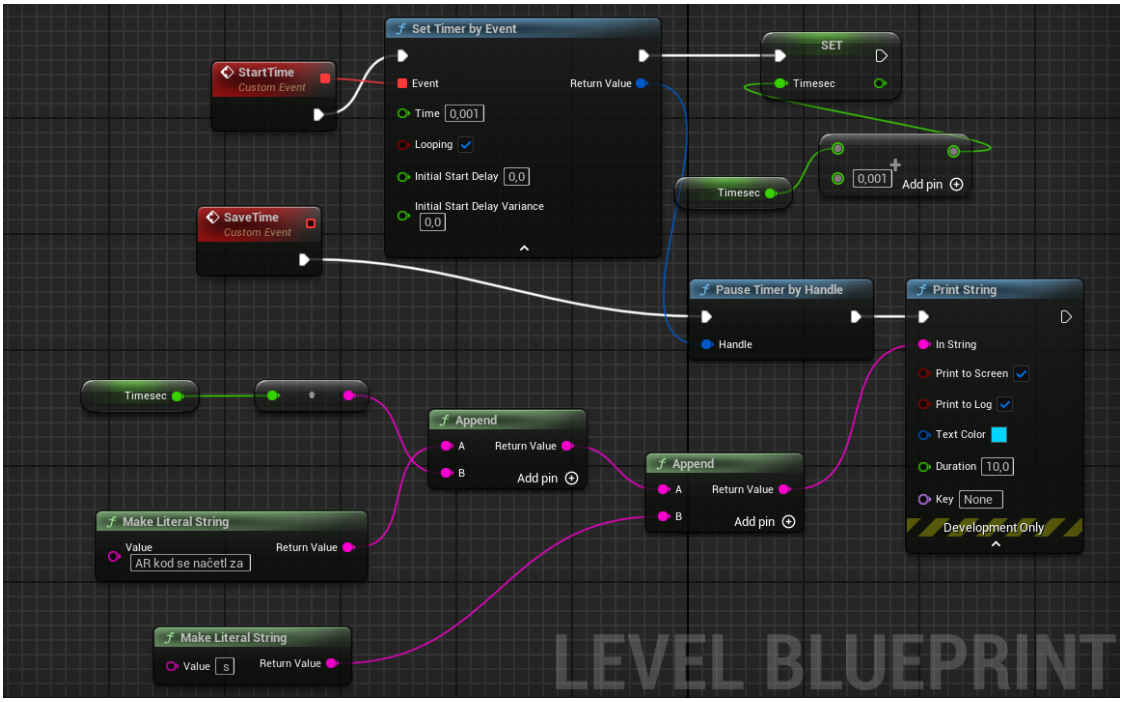
Obrázek 28. Custom event SaveTime

Poté tvoření programu bylo zaměřeno na custom event SaveTime, do Blueprintu byl přidán node Pause Timer by Handle, pomocí kterého se získává čas z návratové hodnoty v node Set Timer by Eventu u node StartTime. Posléze byl přidán node Print String, který slouží k zobrazování stringů a zapisování stringů do logu. Tento node se používá pouze pro vývojové účely. Poté byla nastavena položka Duration na sedm sekund. Posléze byl přidán node To String (Float), pomocí kterého byla převedena proměnná typu float do typu proměnné string. Tohle bylo provedeno z důvodu kompatibility s node Print String, protože proměnou typu float nelze vložit jako vstup do tohoto node. Následně byla přidána do Blueprintu proměnná Timesec. Nakonec nodes byly propojeny mezi sebou podle následujícího obrázku.



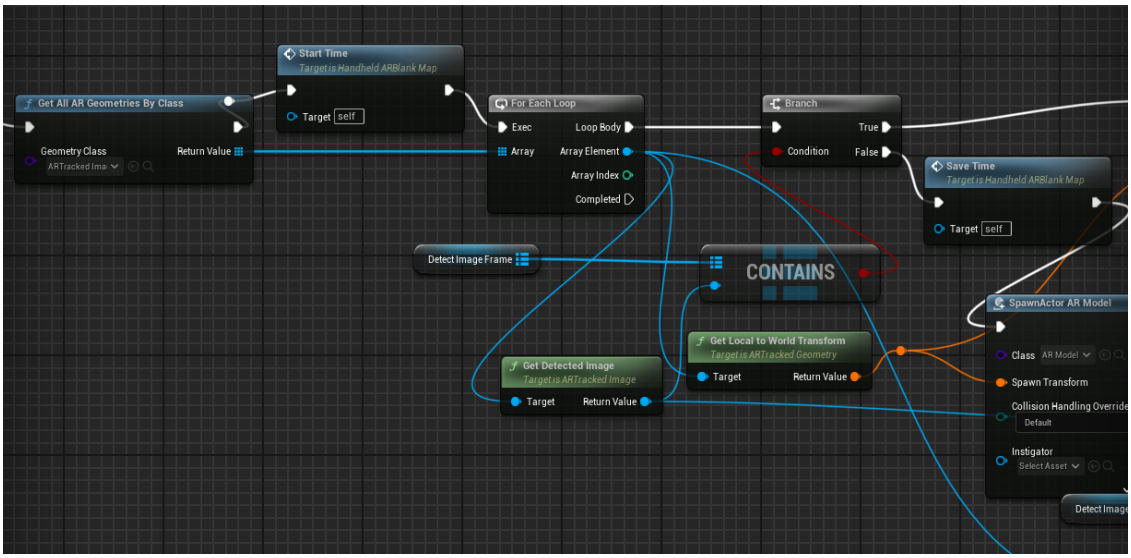
Obrázek 29. Custom event SaveTime

Z toho důvodu, aby se výsledky měření zobrazovaly více uživatelsky přívětivě byly přidány za tímto účelem do Blueprintu dva nodes Make Literal String. Do jednoho z nich bylo napsáno do proměnné text (AR kód se načetl za x sekund) a do druhého bylo napsán string znázorňující sekundy s mezerou před tímhle písmenem. Poté byly přidány dva nodes Append String. Tyto nodes poté byly propojeny tak, aby výsledný string měl předem určený textový formát. Výsledný string byl zapojen do node Print String.



Obrázek 30. Časovač finální

Část týkající časovačů byla dokončena, posléze tyto custom eventy byly vloženy na správné místo do hlavního eventu Event Tick, první event StartTime byl vložen za node Get All AR Geometries by Class z toho důvodu, aby se časovač spustil. Druhý event SaveTime byl vložen za podmínku ve větvi false, který detekuje prvotní přítomnost AR kódu. Tenhle krok znázorňuje obrázek pod tímto textem.



Obrázek 31. Vložení a propojení custom eventů StartTime a SaveTime do finálního Level Blueprint

Tímto byla dokončena část týkající se časovačů. Další podkapitola byla věnována samotnému testování rychlosti načtení všech AR kódu.

6.2 Testování vytvořeného AR kódu a porovnání s ostatními AR kódy

Byly vytisknuty všechny testované AR kódy včetně QR kódu na bílý papír velikosti A4, AR kódy jsou všechny ve stejné velikosti. Mobil byl přichycen na stativ, který byl přesunován v závislosti na požadované vzdálenosti AR kódu od kamery mobilu. Projekt byl vyexportován na mobil s testovací aplikací, kde AR kódy byly připraveny k testování. Místnost byla osvětlena 500 lx.

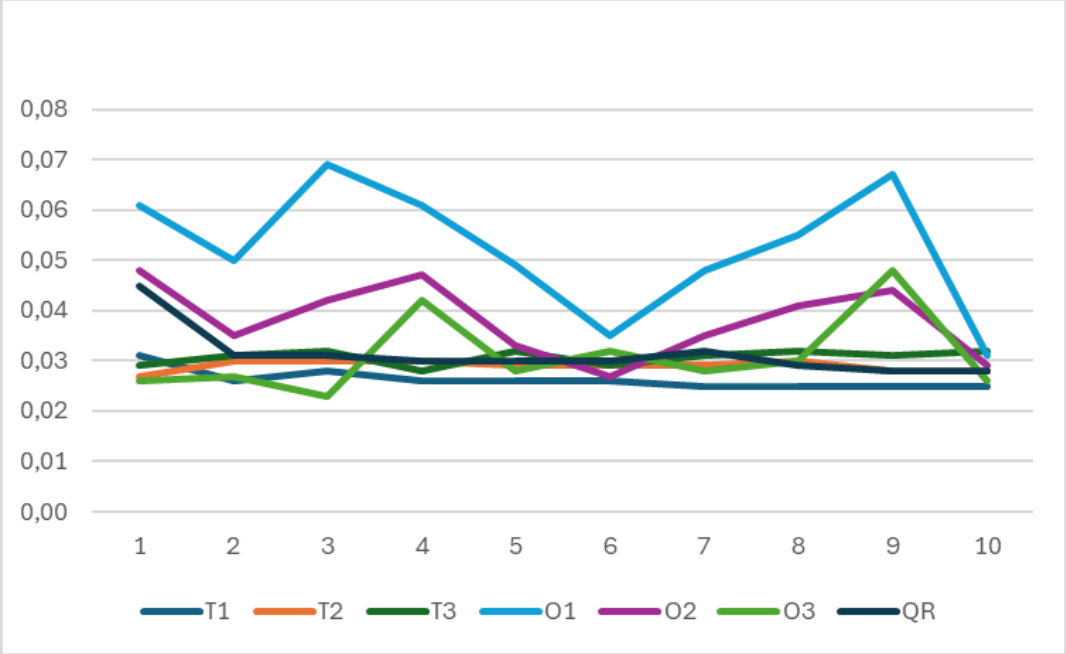
Při prvním měření byl měřen počet sekund AR kódy ze vzdálenosti 15 cm, dokud se nenačetl AR kód z daného obrázku. V tomto měření byl měřen každý kód tímto způsobem desetkrát. Poté byl každý kód desetkrát měřen ze vzdálenosti 30 cm. Z důvodu přehlednosti byly kódy pojmenovány jako T1 až T3 a O1 až O3. Písmeno T označuje Tritový kód a písmeno O naopak kód obrázkový. Z nichž první varianta je neupravovaný AR kód. Druhá varianta je pixelizovaný AR kód 16 bloky. Třetí varianta je pixelizovaný AR kód 32 bloky. Následně po všech měřeních byl vytvořen graf zobrazující všechny vypočítané průměry rychlostí načtení AR kódů. Pro porovnání byl také naměřen všeobecně využívaný QR kód za stejných podmínek. Následující tabulka toto označení popisuje.

Tabulka 1. Označení kódů

AR kódy	Označení v tabulce
Tritový kód	T1
Tritový kód s pixelizací 16 bloků	T2
Tritový kód s pixelizací 32 bloků	T3
Obrázkový kod	O1
Obrázkový kod s pixelizací 16 bloků	O2
Obrázkový kod s pixelizací 32 bloků	O3
QR kód	QR

Tabulka 2. Měření rychlosti načtení AR kódů ze vzdálenosti 15 cm

	Měření [s]									
AR Kódy	1	2	3	4	5	6	7	8	9	10
T1	0,031	0,026	0,028	0,026	0,026	0,026	0,025	0,025	0,025	0,025
T2	0,027	0,030	0,030	0,030	0,029	0,029	0,029	0,030	0,028	0,028
T3	0,029	0,031	0,032	0,028	0,032	0,029	0,031	0,032	0,031	0,032
O1	0,061	0,050	0,069	0,061	0,049	0,035	0,048	0,055	0,067	0,031
O2	0,048	0,035	0,042	0,047	0,033	0,027	0,035	0,041	0,044	0,029
O3	0,026	0,027	0,023	0,042	0,028	0,032	0,028	0,030	0,048	0,026
QR	0,045	0,031	0,031	0,030	0,030	0,030	0,032	0,029	0,028	0,028



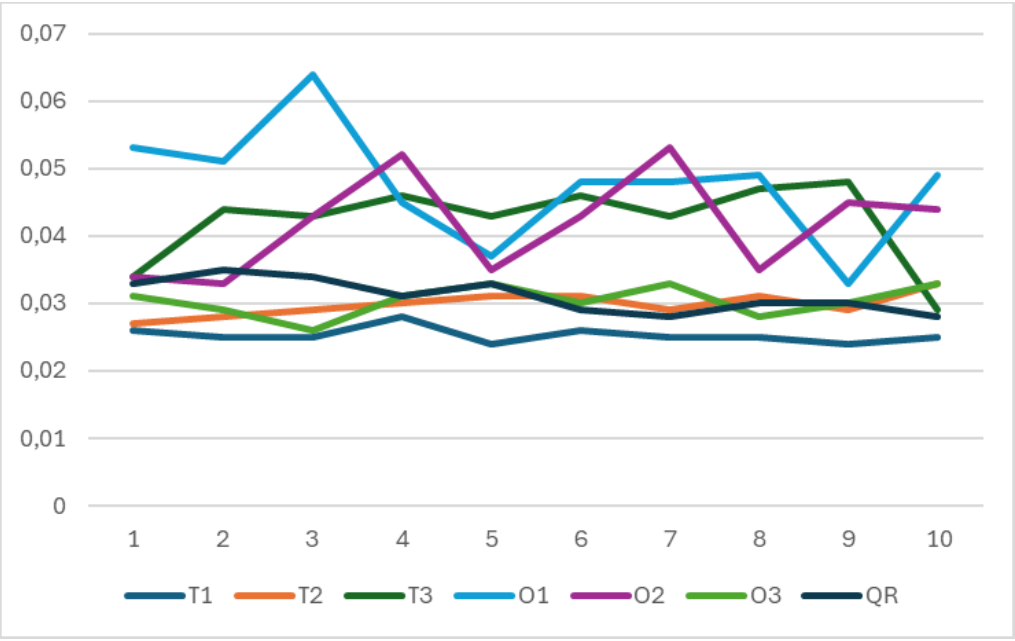
Obrázek 32. Měření rychlosti načtení AR kódů ze vzdálenosti 15 cm

V uvedeném grafu lze spatřit několik věcí, první z nich je Tritový kód 1 a 2 spolu s obrázkovým kódem 3 vycházejí podle měření lepší výsledný čas rychlosti načtení AR kódu čtečkou kódů. Tyto tři kódy zároveň vykazují nižší volatilitu výsledku rychlosti načtení AR kódu, tedy větší spolehlivost. Zároveň byl potvrzen předpoklad, že AR kód po pixelizaci,

která v konkrétním případě ubírá bloky pixelů zvyšuje jednoznačně rychlost načtení AR kódu.

Tabulka 3. Měření rychlosti načtení AR kódů ze vzdálenosti 30 cm

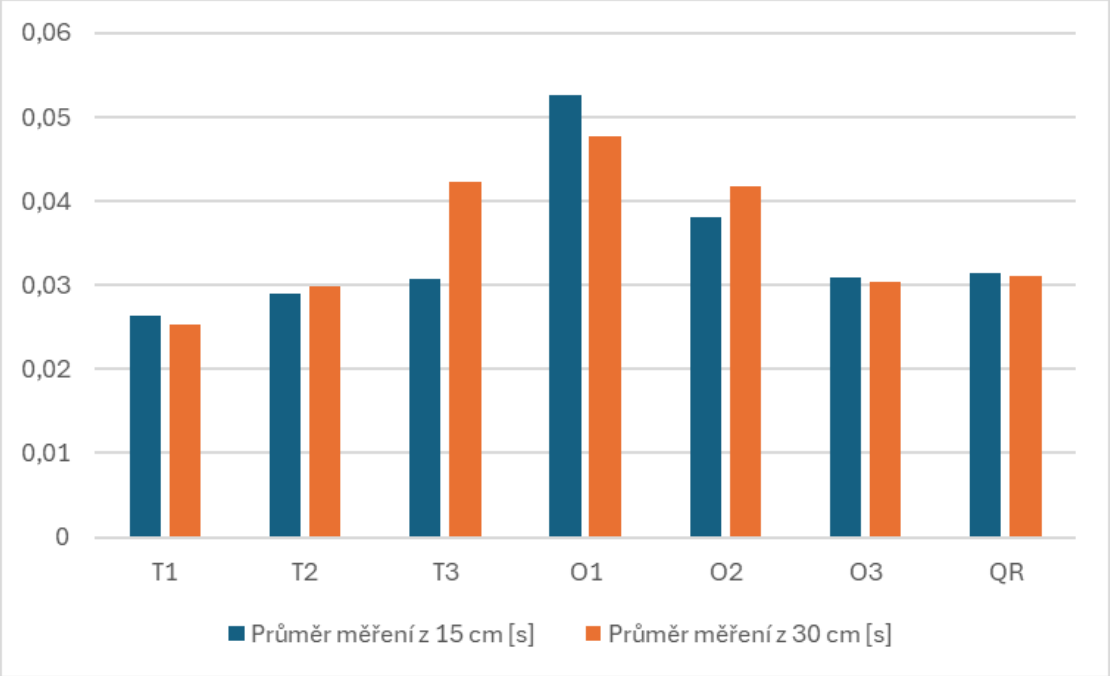
	Měření [s]									
AR Kódy	1	2	3	4	5	6	7	8	9	10
T1	0,026	0,025	0,025	0,028	0,024	0,026	0,025	0,025	0,024	0,025
T2	0,027	0,028	0,029	0,030	0,031	0,031	0,029	0,031	0,029	0,033
T3	0,034	0,044	0,043	0,046	0,043	0,046	0,043	0,047	0,048	0,029
O1	0,053	0,051	0,064	0,045	0,037	0,048	0,048	0,049	0,033	0,049
O2	0,034	0,033	0,043	0,052	0,035	0,043	0,053	0,035	0,045	0,044
O3	0,031	0,029	0,026	0,031	0,033	0,030	0,033	0,028	0,030	0,033
QR	0,033	0,035	0,034	0,031	0,033	0,029	0.028	0,030	0,030	0,028



Obrázek 33. Měření rychlosti načtení AR kódů ze vzdálenosti 30 cm

Tabulka 4. Průměr rychlosti načtených AR kódů

AR Kódy	Průměr měření z 15 cm [s]	Průměr měření z 30 cm [s]
T1	0,0263	0,0253
T2	0,0290	0,0298
T3	0,0307	0,0423
O1	0,0526	0,0477
O2	0,0381	0,0417
O3	0,0310	0,0304
QR	0,0314	0,0311



Obrázek 34. Průměr rychlosti načtených AR kódů

Z posledního grafu bylo zjištěno, že Tritový kód T1 a T2 společně s obrázkovým kódem O3 mají lepší rychlost načtení AR kódů čtečkou kódů než klasický QR kód. Z grafu je také zřejmě vidět, že vzdálenost AR kódu od čtecího zařízení má u některých kódu vliv na celkovou rychlost načtení AR kódu. Jednoznačně nejhorší výsledky měl Obrázkový kód O1, nejlepší výsledky naopak měl Tritový kód T1. Největší rozdíl z hlediska obou vzdáleností měření měl jednoznačné Obrázkový kód, který měl horší výsledky měření ze vzdálenosti 30 cm. Bylo celkem provedeno 140 měření. Všechny cíle zadání byly splněny.

ZÁVĚR

V teoretické části byly popsány základní vlastnosti a využití rozšířené reality. Následně byly zmíněny standardy rozšířené reality, které se aktuálně využívají při vývoji aplikací s rozšířenou realitou. Poté byly popsány základní funkce programu Unreal Engine.

V praktické části byly vytvořeny dva vlastní AR kódy, které měly překonat vlastnosti současně dostupných AR kódů, jako první byl vytvořen Tritový kód, který byl zaměřen na co nejrychlejším načtení čtecím zařízením, jako druhý byl vytvořen Obrázkový kód, který se zaměřil na vizuální stránku kódu. Následně byly pomocí programu GIMP vytvořeny zpixelizováné alternativy obou vytvořených vlastních AR kódů, ty měly za cíl zjistit vliv počtu pixelů na celkovou rychlost načtení AR kódu. V následující kapitole byl pomocí herního enginu Unrel Engine 5.3.2 vytvořen pomocí blueprintů program pro implementaci vlastního kódu AR pro skenování a vkládání dodatečného obsahu. Poté byl tento program vyexportován na mobilní telefon pro otestování základních funkcí.

Do programu byla následně vytvořena metoda pro měření rychlosti načtení AR kódu čtecím zařízením pomocí časovače, který se zastavil při načtení AR kódu a zobrazil na obrazovce mobilního zařízení výsledný počet sekund. Poté bylo u každého AR kódu včetně všeobecně užívaného QR kódu provedeno deset měření ze vzdálenosti 15 cm a deset měření ze vzdálenosti 30 cm. Tyto měření potvrdily, že při pixelizaci, při které se zmenšují pixely dochází k zrychlování načtení AR kódu čtecím zařízením. Z hlediska rychlosti načtení AR kódu byl QR kód překonán Tritovým kódem T1 a T2 společně s Obrázkovým kódem O3. Celkem bylo provedeno 140 měření. Finální program je k dispozici na vyžádání u autora práce na mailu: a_kopriva@utb.cz.

Všechny cíle práce byly splněny. Celkovým záměrem bakalářské práce bylo přispět k dalšímu rozvoji a porozumění oblasti AR kódů, a zároveň poskytnout konkrétní praktický přínos pro vývojáře a uživatele v této oblasti.

SEZNAM POUŽITÉ LITERATURY

[1] *What is augmented reality (AR)?* [online]. [cit. 2024-03-17]. Dostupné z: https://www.sap.com/products/scm/industry-4-0/what-is-augmented-reality.html

[2] What *is Virtual Reality?* [online]. [cit. 2024-03-17]. Dostupné z: https://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html

[3] *Co je hybridní realita?* [online]. [cit. 2024-03-17]. Dostupné z: https://learn.microsoft.com/cs-cz/windows/mixed-reality/discover/mixed-reality

[4] ONG, Soh Khim, YEH CHING NEE, Andrew, ed., 2023. *Handbook of Augmented Reality*. Springer. ISBN 978-3-030-67821-0.

[5] *The History of Virtual Worlds* [online]. [cit. 2024-03-17]. Dostupné z: https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/VirtualWorlds/styled/page1.html

[6] *Overview of ARCore and supported development environments* [online]. [cit. 2024-03-17]. Dostupné z: https://developers.google.com/ar/develop

[7] *Introduction to ARKit* [online]. [cit. 2024-03-17]. Dostupné z: https://designcode.io/arkit-intro

[8] *OpenXR* [online]. [cit. 2024-03-17]. Dostupné z: https://learn.microsoft.com/cs-cz/windows/mixed-reality/develop/native/openxr

[9] *Augmented Reality in Education* [online]. [cit. 2024-03-17]. Dostupné z: https://www.apple.com/education/docs/ar-in-edu-lesson-ideas.pdf

[10] *Microsoft HoloLens 2 in Medical and Healthcare Context* [online]. [cit. 2024-03-17]. Dostupné z: https://encyclopedia.pub/entry/33528

[11] *Kód rozšířené reality* [online]. [cit. 2024-03-17]. Dostupné z: https://ar-code.com/cs

[12] *History of QR Code* [online]. [cit. 2024-03-17]. Dostupné z: https://www.qrcode.com/en/history/

[13] *And That's How QR Codes Work?* [online], 2020. [cit. 2024-03-31]. Dostupné z: https://www.sproutqr.com/blog/how-do-qr-codes-work

[14] *SQRC®* [online]. [cit. 2024-03-17]. Dostupné z: https://www.denso-wave.com/en/system/qr/product/sqrc.html

[15] *FrameQR Codes: A New Way To Design QR Codes* [online]. [cit. 2024-03-17]. Dostupné z: https://qrcodecreator.com/qr-code-frame

[16] *What is ArU-code?* [online]. [cit. 2024-03-31]. Dostupné z: https://www.womnet.com/aru-code/

[17] *Unreal Engine: Overview, Features and Benefits* [online]. [cit. 2024-03-17]. Dostupné z: https://juegostudio.medium.com/unreal-engine-overview-features-and-benefits-728021ce6542

[18] *Programming and Scripting* [online]. [cit. 2024-03-17]. Dostupné z: https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/

[19] NIXON, David, 2020. *Beginning Unreal Game Development*. Apress. ISBN 978-1-4842-5638-1.

[20] *Designing Visuals, Rendering, and Graphics* [online]. [cit. 2024-04-03]. Dostupné z: https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/

[21] COOKSON, Aram, Ryan DOWLINGSOKA a Clinton CRUMPLER, 2016. *Sams Teach Yourself Unreal® Engine 4 Game Development in 24 Hours*. Pearson Education. ISBN 978-0-672-33762-8.

[22] VALCASARA, Nicola, 2015. *Unreal Engine Game Development Blueprints*. Packt Publishing. ISBN 978-1-78439-777-7.

[23] *AR development in Unity* [online], 2024. [cit. 2024-04-03]. Dostupné z: https://docs.unity3d.com/Manual/AROverview.html

[24] *Godot Engine 3.5 documentation in English* [online]. [cit. 2024-04-03]. Dostupné z: https://docs.godotengine.org/en/3.5/tutorials/vr/xr_primer.html

[25] *REAL TIME DISTANCE CALCULATION USING ARUCO MARKERS* [online], 2021. [cit. 2024-04-02]. Dostupné z: https://medium.com/analytics-vidhya/real-time-distance-calculation-using-aruco-markers-b469d5f9791d

[26] *Augmented Reality Is Used in Minimally Invasive Spine Surger* [online], 2020. [cit. 2024-03-29]. Dostupné z: https://www.engineering.com/story/augmented-reality-is-used-in-minimally-invasive-spine-surgery

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: LEVEL BLUEPRINT