

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

**Prototyp klientské aplikace pro
komunitní překlad textů z kulturních
institucí**

Plzeň 2022

Jan Pelikán

Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení: **Jan PELIKÁN**
Osobní číslo: **A19B0157P**
Adresa: **Blatenská 28, Plzeň – Lobzy, 32600 Plzeň 26, Česká republika**
Téma práce: **Prototyp klientské aplikace pro komunitní překlad textů z kulturních institucí**
Téma práce anglicky: **Prototype of client application for the community translation system of the cultural institution texts**
Vedoucí práce: **Ing. Richard Lipka, Ph.D.**
Katedra informatiky a výpočetní techniky

Zásady pro vypracování:

- Seznamte se se stávajícími technologiemi a aplikacemi mobilních průvodců a poskytovanými službami.
- Seznamte se s problematikou komunitního překladu.
- Navrhněte prototyp klienta mobilního průvodce s možností získávání dat z kulturních institucí.
- Navržený prototyp implementujte pro vybranou platformu.
- Vytvořenou implementaci otestujte.

Seznam doporučené literatury:

Dodá vedoucí práce.

Podpis studenta:

Datum:

Podpis vedoucího práce:

Datum:

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 15.10.2021

Jan Pelikán

Abstract

Abstrakt

Obsah

1	Úvod.....	6
2	Dostupné technologie pro mobilní aplikace a jejich vývoj	7
2.1	Augmented reality.....	7
2.1.1	Knihovna ARCore.....	7
2.1.2	Knihovna ARKit.....	8
2.1.3	Wikitude	8
2.1.4	Shrnutí	8
2.2	QR kód.....	8
2.2.1	Knihovna Barcode scanner.....	8
2.2.2	Knihovna Code scanner.....	9
2.2.3	Knihovna MobileVisionBarcodeScanner	9
2.2.4	Porovnání s čárovým kódem	9
2.2.4	Shrnutí	9
2.3	Text to speech	9
2.3.1	Voice RSS	10
2.3.2	Shrnutí	10
2.4	NFC tagy	11
2.4.1	Shrnutí	11
2.5	GPS	11
2.6	Překlad textů	13
2.7	Programovací jazyk	13

1 Úvod

2 Dostupné technologie pro mobilní aplikace a jejich vývoj

Tato kapitola se bude zabývat analýzou dostupných knihoven, které by mohli sloužit pro vývoj interaktivního mobilního průvodce. Díky této kapitole získáme přehled o volně dostupných funkcionalitách, které případně lze využít při vývoji aplikace. Daná znalost nám pomůže při návrhu a programování průvodce.

2.1 Augmented reality

Augmented reality (AR), nebo také česky Rozšířená realita je v dnešní době často využívanou technologií na poli mobilních zařízení. Pojďme si rozšířenou realitu blíže představit. O co se vlastně jedná a jaké má využití v mobilních aplikacích. Rozšířená realita je způsob zobrazení virtuálně vytvořených objektů do reálného prostoru. Pod virtuálními objekty si můžeme představit pohyblivé i nehybné postavy případně i „vylepšené“ objekty z reálného světa, může se jednat i o audio nebo pouze o fotky. Vytvoření těchto objektů závisí na podnětu z reálného světa. Mobilní fotoaparát nasnímá reálný objekt, který slouží jako podnět právě pro vytvoření virtuálního objektu.

Dnes můžeme vidět rozšířenou realitu běžně využitou například v edukativních aplikacích (Experience Real History -> využití obrázků a virtuálních karet k zlepšení znalostí z historie), hrách (Pokémon GO -> možnost zobrazení pokémonů v reálném prostředí) nebo na sociálních sítích (Facebook, Instagram nebo Snapchat -> využívají filtry pro vylepšení fotek uživatelů).

Ovšem aby vylepšená realita správně fungovala, je potřeba aby zařízení splňovalo minimální požadavky. Mobilní zařízení musí podporovat lokační služby (GPS), fotoaparát a některé aplikace mohou vyžadovat připojení k internetu. Zároveň mobilní telefon musí mít dostatečný výkon, aby byl schopný generovat a využít rozšířenou realitu.

Nyní bych rád přiblížil mnou vybrané knihovny pro AR:

2.1.1 Knihovna ARCore

Jedná se o velkou knihovnu od firmy Google, je dostupná zdarma v celém rozsahu. Minimální požadavky jsou Android verze 7.0 a vyšší a API level 24. Díky tomuto požadavku je tato knihovna spustitelná na 73,7% zařízení s Android operačním systémem [7]. Knihovna má širokou škálu funkcionalit jako například: nastavení správného světla přidaného objektu, tak aby správně „zapadl“ do reality, vylepšit (rozpohybovat) obrazy i obličej a zajištění správného postavení objektu, to znamená objekt, který má být v pozadí je překrýván tím v popředí. Knihovna nepodporuje rozpoznávání 3D objektů, ovšem na GitHubu už je toto téma v issues [1]. Díky tomu, že je tato knihovna od Googlu, je zde obsáhlá dokumentace i se zdrojovými kódy pro Android (Java/Kotlin).

2.1.2 Knihovna ARKit

Další velká knihovna pro rozšířenou realitu. Obsahuje dosti podobné funkce jako předešlá knihovna ARCore, knihovna je obohacena o práci s 3D objekty a jejich rozpoznáváním, kterým zatím knihovna ARCore nedisponuje. 3D objekty by se v našem případě hodily, protože bychom mohli potřebovat skenovat například sochu či **bistu**. Knihovna je vytvořená a vylepšovaná firmou Apple, obsahuje rozsáhlou dokumentaci obohacenou o „best practices“ použití. Tím, že je knihovna od Apple, funguje pouze pro iOS a iPadOS.

2.1.3 Wikitude

Jedná se o all-in-one framework, který podporuje jak Android, tak iOS. Je to hlavní produkt firmy Wikitude. Má stejné funkcionality jako předešlé dvě knihovny, k tomu přidává své další, kterými jsou: Geolokace uživatele, implementace 3D objektů i s jejich skenováním a dokáže sledovat a více objektů zároveň a reagovat na jejich změny. Cena licence začíná od 2490 eur pro jedno použití nebo roční předplatné za 2990 eur [2].

2.1.4 Shrnutí

Z výše vypsanych knihoven, jejich funkcí a ceny **vyplývá**, že pro implementaci rozšířené reality (AR) by jedinou vhodnou knihovnou pro náš případ bylo ARCore. Bohužel ARCore neumí **rozpoznávat 3D modely**, tudíž by muselo být „obohaceno“ o nějaký skener, který by dokázal rozpoznávat objekty. Implementace a spojení skeneru a ARCore by bylo složité a aplikaci by to z mého **pohledu tolik nevylepšilo**. Zároveň by se musela zajistit komunikace mezi **serverem, skenerem a ARCore**. Možným využitím rozšířené reality by bylo, že data, která by přišla ze serveru (například po naskenování QR kódu) by se zobrazila jako text/fotka do volného prostoru vedle exponátu, tuto funkcionalitu ARCore umí a dal by se případně k tomuto využít.

2.2 QR kód

QR kód je **nově moderní** technologií pro předávání dat v jednoduché a dostupné formě. Kód může naskenovat a použít kdokoli, kdo vlastní chytrý mobilní telefon s fotoaparátem pro skenování. Jejich pomocí se dá vytvořit odkaz na webovou stránku nebo předat pouze jen text. Do QR kódu se dá zakódovat až 4296 alfanumerických znaků případně 7089 čistě numerických znaků [3].

QR kódy by se v našem případě dali využít tím způsobem, že u každého exponátu by se nacházel jeden QR kód, který by se dal snadno a rychle naskenovat. Výsledkem skenu by mohl být nějaký kód, díky kterému by server jednoznačně věděl, **o který exponát se jedná**. Mohl by díky tomu rychle a vždy správně odeslat uživateli data o exponátu, nejspíše se bude jednat o text. Vytvoření QR kódu jako takového nic nestojí, jediné náklady by byli na vytisknutí QR kódů.

Chtěl bych vypsát některé dostupné knihovny, které by se dali využít pro zpracování QR kódu. Většina knihoven, co jsem našel, se nacházejí pouze na GitHubu, a proto k nim je pouze limitované množství informací:

2.2.1 Knihovna Barcode scanner

Jedná se o nejstarší knihovnu, která byla vyvíjena od roku 2013. Základ knihovny je postaven na **ZXing** („Zebra Crossing“) knihovně. Z této knihovny vychází, mnoho nových knihoven pro čtení QR kódu či čárových kódů. Podporuje pouze základní

funkcionalitu. Od roku 2020 již není udržovaná a updatovaná. Odkaz na Github s knihovnou [4].

2.2.2 Knihovna Code scanner

Základ knihovny je postaven na předešlé knihovně [ZXing](#). Tato knihovna navíc podporuje přední a zadní kameru, přizpůsobitelný snímač QR kódu, orientaci kamery na výšku a šířku a autofocus. Funguje pro Android API 19+, verze 4.4 KitKat. Díky tomuto se dá knihovna využít na 98,1% zařízení. V popisu knihovny je detailní popis použití, jak v Kotlinu, tak i pro Javu. Odkaz na Github [5].

2.2.3 Knihovna MobileVisionBarcodeScanner

Tato knihovna je hodně podobná té předešlé (Code scanner). Hlavním rozdílem je, že dovoluje uživateli více si **konfigurovat snímač kódu**. Samozřejmě pokud programátor tyto funkce nezakáže. Podporuje autofocus, vykresluje čtverec kolem aktuálně skenovaného kódu. Odkaz na knihovnu [6].

2.2.4 Porovnání s čárovým kódem

Srovnání s kódem čárovým typu EAN-13 případně EAN-8. QR kód nám dokáže zakódovat mnohem více informací než standardní čárový kód, jenže v našem případě nám stačit pouze identifikační číslo výrobku, které se odešle na server a v databázi se exponát vyhledá. V našem případě se do čárového kódu typu EAN-8 dá zakódovat 7 číslic, které jednoznačně určí exponát, to pro náš **případ je dostatečné**.

Na druhou stranu, skenování QR kódu je pro uživatele jednodušší, protože se dá skenovat například i vzhůru nohama. To znamená, že po uživateli nechceme, aby musel otáčet telefon do různých úhlů. Zároveň je v dnešní době používání QR kódu moderní, a většině uživatelům tak rychle dojde, že zrovna toto mají naskenovat.

2.2.4 Shrnutí

QR kód by z mého pohledu mohl být dobrým nástrojem pro použití, při implementaci mobilního průvodce pro jednoznačné určení exponátu. Do QR kódu by se dal zakódovat speciální řetězec, který by serveru jednoznačně určil, o který exponát se jedná. Všechny z výše vypsanych knihoven by se daly pro realizaci využít. Osobně jsem si práci s těmito knihovnami zkoušel a nejlépe se mi pracovalo s druhou knihovnou (Code scanner).

V porovnání s čárovým kódem si myslím, že z pohledu uživatele bude více „user friendly“ použití QR kódu, v dnešní době jsou QR kódy na hodně místech použity a uživatel si rychle spojí, že QR kód má naskenovat. Výsledek bude záviset na preferenci Západočeského muzea.

2.3 Text to speech

Česky lze říkat „Text na řeč“ nebo lépe „Syntéza řeči“. Programy používané pro syntézu řeči produkují slova, v lepším případě celé věty přesně tak, jak je psané v textu, který čtou. Využívají k tomu databáze, kde jsou uloženy fóny, difóny nebo celá slova. Díky této interpretaci program dokáže poskládat celá slova a věty v jeden celek.

Fón neboli hláska je základním prvkem mluvené řeči. Tvorbou a fyzikálními vlastnostmi fónu se zabývá vědní disciplína fonetika. Difón je část zvukové informace

skládající se ze dvou sousedních fonémů. Laicky by se dalo říct, že fón je písmeno a difón je slabika.

Implementovat funkci text to speech do mobilního průvodce by šlo tím způsobem, že by uživateli byla nabídnuta možnost, zda daný přeložený text chce přečíst. Ze serveru přijde odpověď v podobě textu k danému exponátu, uživatele bychom se zeptali, zda se chce daný přeložený text přečíst sám, nebo pokud má sluchátka, zda text chce přečíst pomocí „aplikace“.

Podstatným bodem text-to-speech (TTS) u naší aplikace může být potřeba aby více uživatelů v jeden moment využilo tuto funkci. Dobrým způsobem, jak toto vyřešit by byl offline TTS, každému uživateli by se na vlastním telefonu vytvořil zvuk a nemusel by odesílat žádný požadavek pro službu TTS. Ovšem, nepodařilo se mi najít žádnou knihovnu, která by toto podporovala kromě již implementované třídy `TextToSpeech`, která je implicitně v android telefonech. Knihovna nepodporuje český jazyk, proto by funkce pro češtinu nemohla fungovat. Pro cizince už by TTS mohlo fungovat, protože knihovna podporuje angličtinu, francouzštinu, němčinu, italštinu a španělštinu.

Druhým možným způsobem by bylo využít službu TTS online, tím způsobem, že by uživatel obdržel text ze serveru, a poté by odeslal požadavek na službu TTS, že chce daný text přečíst. Problémem tohoto řešení je možná zátěž serveru. V případě, že přijede zájezd cizinců a všichni budou chtít využít tuto službu v cizím jazyce, bude se jednat o desítky požadavků během minuty. Tento případ je pravděpodobný v muzeích a galeriích a pro server není zanedbatelný. Všechny služby podporující online požadavky pro službu TTS mají zpoplatněny zpracování většího objemu dat. Zmíním pouze jednu knihovnu, která zdarma nabízí **nejvíce služeb**.

2.3.1 Voice RSS

Služba od této společnosti nijak nelimituje počet požadavků za minutu, nebo nelimituje počet uživatelů. Jediným měřítkem je počet požadavků za den, kterým je 350 ve verzi, která je zdarma. Toto číslo není úplně malé a v raných verzích aplikace by určitě stačilo, ale nemůžeme se na toto spolehnout.

Služba nabízí TTS SDK pro Android i jiné jazyky, a komunikuje dále přes webové protokoly HTTP GET a HTTP POST. Nabízí přes 45 světových jazyků včetně češtiny, němčiny a angličtiny. Více možných hlasů u každého z jazyků.

2.3.2 Shrnutí

Funkcionalita text-to-speech by byla uživatelsky přívětivá a dle mého i oblíbená, protože již v dnešní době existují pro muzeích a památkách audio průvodci. Tato featurea by mohla z naší aplikace udělat „volnějšího“ audio průvodce, pokud by o to měl uživatel zájem. Výhodou by bylo více možnost, jak možnost čtení textu samotným uživatelem, tak možnost TTS.

Pokud by se funkce TTS implementovala do aplikace, s největší pravděpodobností bude knihovna Androidu. Funkce by fungovala pouze pro turisty, kteří nemluví českým jazykem.

Jinou možností, jak využít text-to-speech by bylo pouhé doporučení již předem vybrané aplikace, kterou by si uživatel stáhl. Naše aplikace by už jen text k přečtení zkopírovala do schránky a poté by si jen uživatel daný text vložil do aplikace a nechal přečíst.

2.4 NFC tagy

Near field communication je technologie pro bezdrátový přenos dat mezi elektrickými zařízeními na krátkou vzdálenost. V dnešní době se využívá například pro bezkontaktní platební terminály. NFC může zprostředkovat obousměrnou výměnu dat, či pouze jednosměrnou. Při použití jednosměrného přenosu dat, se říká že se jedná o pasivní NFC „tag“.

Pro náš účel by byl vhodný právě tento jednosměrný způsob komunikace. NFC tag by sloužil jako identifikátor daného exponátu podobně jako QR kód či čárový kód. Při přiložení telefonního zařízení k NFC tagu by se předal požadavek na server, který by podle identifikátoru věděl, který text z databáze má vrátit.

Výhody NFC tagu:

Přepisovatelnost – data uložená v NFC tagu je možné **kdykoli přepsat**. Tato funkce se dá využít třeba u expozic, které mají omezenou dobu trvání. NFC tag by se jednoduše po ukončení expozice přepsal a mohl by být přesunut k jinému exponátu. Naproti tomu u QR kódu/čárového kódu tato možnost není a byla by potřebovat vytvořit kód nový.

Skenování – v některých situacích může být pro uživatele jednodušší pouze telefon přiložit k zařízení než správně zaměřit fotoaparát a čekat, než se naskenuje kód. Pro starší uživatele aplikace to může být celkově jednodušší či pokud uživatel nemůže zrovna využít druhou ruku, aby skenování proběhlo jednoduše.

Nevýhody NFC tagu:

Cena – cena jednoho NFC tagu se pohybuje **od 15 Kč. (15.11.2021)**. Tento NFC tag je nepřepsatelný a svojí hodnotu má již v sobě vloženou od výrobce. Oproti tomu jediný náklad pro QR/čárový kód je papír a tiskárna, přičemž na jeden papír se dá vytisknout více kódů. Z toho vyplývá, že výroba kódu je levnější.

Vytvoření – připravení NFC tagu jako takové je značně obtížnější než třeba QR/čárový kód. Pokud by se zvolil způsob pomocí NFC tagů, samotný tag se musí objednat/koupit, čekat, než tag přijde, a následně do něj zakódovat identifikátor pro server. Při použití QR kódu stačí lze využít internetové generátory, kam se zadá identifikátor a QR kód se vygeneruje a následně jen vytiskne. Tento způsob je méně náročný a rychlejší.

2.4.1 Shrnutí

Použití NFC tagu by bylo výhodné, pokud by muzeum často měnilo expozice a exponáty. Díky tomu, že NFC tag se dá jednoduše přepsat, bylo by to jednodušší než se starat o nové tisknutí kódů. Na druhou stranu, připravit a vytisknout kód není o tolik náročnější než přepsat NFC tag. Při použití NFC tagu by nebyla potřeba implementovat čtečku kódů do aplikace.

2.5 GPS

GPS neboli globální polohový systém je technologie, která umožňuje pomocí elektronického přijímače určit přesnou polohu na povrchu země. Celá technologie funguje na principu přijímání signálu. Elektronické zařízení (mobil, navigace) obsahuje přijímač, který přijímá signál z družic, které obíhají zemi, díky tomu družice dokáží vypočítat pozici přijímače na jednotky metrů.

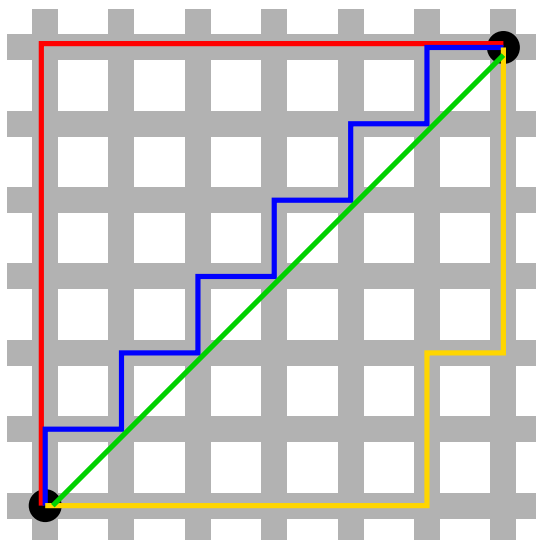
Pro naši aplikaci by se tento systém dal využít k návrhu muzeí, ve kterém se uživatel nachází. Uživatel bude chtít vybrat ve kterém se nachází muzeu, při tomto požadavku by bylo vhodné seřadit muzea ve vzdálenosti od uživatele od nejbližšího muzea po nejvzdálenější. Tato funkce by ulehčila hledání muzea, protože hledané muzeum by bylo vždy mezi prvními výsledky.

Pokud bude v databázi muzeí málo, můžeme počítat vzdálenost pro každé muzeum při každém požadavku. Ovšem až se databáze rozroste, tento požadavek bude dlouhý, bude se jednat o sekundy až desítky sekund. To pro uživatele není dobré. Níže bych chtěl popsat, jakým způsobem lze tento problém řešit.

Možnosti výpočtu vzdálenosti:

Euklidovská vzdálenost – Jedná se o vzdálenost, kterou bychom mohli nazvat také vzdušnou čarou, jedná se o nejkratší cestu z jednoho bodu do druhého. V prostoru je tato vzdálenost počítána jako odmocnina ze sumy rozdílů druhých mocnin každé souřadnice zvlášť. Pro lepší představu, máme dva body A a B, každý z nich má souřadnici x a y. Obecný vzorec by vypadal takto: $m_e(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$. Prakticky by proběhli dva výpočty, jednou pro souřadnici x a po druhé pro souřadnici y. Tento způsob je výpočetně složitý, protože se počítají druhé mocniny a zároveň odmocniny.

Manhattan vzdálenost – Jméno této vzdálenosti je odvozeno od ostrova Manhattan v New Yorku, kde většina ulic je k sobě kolmá a tvoří takovou mříž. Tato vzdálenost je počítána jako suma absolutní hodnota rozdílu dvou souřadnic. Pokud bychom měli znovu body A a B se souřadnicemi x a y. Obecný vzorec by vypadal takto: $m_m(A, B) = \sum_{i=1}^n |a_i - b_i|$. Pro výpočet této vzdálenosti je potřeba pouze absolutní hodnota a odčítání/sčítání. Proto pro velké množství dat, je tento způsob mnohem rychlejší. Na druhou stranu, je také tento způsob **více nepřesný**. Na obrázku číslo 1, lze vidět zelenou barvou nakreslenou euklidovskou vzdálenost o délce přibližně 8,49. Modrá, červená a žlutá jsou vše Manhattenská vzdálenosti a mají délku 12.



1 - Ilustrativní obrázek [8]

Jiným, již hotovým řešením se zabývají na následujícím odkaze [9]. Je zde návod, jak využít MySQL příkazy, pro zjištění vzdáleností od bodu v kilometrech. Jedná se o kruhovou vzdálenost od bodu na mapě. Pro nás by bodem na mapě byla poloha uživatele a v okruhu bychom hledali muzea či galerie. V tomto případě je pro získání údajů využít

pouze jeden příkaz, který zredukuje počet dat přenášených z databáze. Vzdálenost počítají přímo v příkaze, tudíž eliminují velkou část dat, která by se přenášela zbytečně, pokud bychom vzdálenost počítali mimo databázi.

2.6 Překlad textů

Tento odstavec se bude zabývat možným strojovým překladem ještě nepřeložených textů, jednalo by se spíše o vylepšení pro server, ale nápad pochází z mé hlavy, proto ho zmíním. V případě, že by si uživatel zažádal o překlad do jazyka, pro který ještě přeložený text nemáme, server by v tento moment poslal požadavek na překladač (možnost využít DeepL překladač, má vytvořené API rozhraní a zároveň hezké strojové překlady) o překlad daného textu, přeložený text by odeslal uživateli a zároveň by si text uložil do databáze jako překlad. Přidal by se například flag „strojový překlad“, aby bylo jasné, že je stále potřeba udělat překlad, ale již nějaký překlad existuje. Využitím tohoto „on-demand“ přístupu by server nebyl nucený často využívat služby překladače a zároveň měl i překlad pro každého uživatele. Implementace závisí hlavně na programátorovi serveru, tudíž on rozhodne, zda se využije mnou navržený způsob. Jiný způsob, na kterém jsme se dohodli, již dříve, byl takový, že uživatel si zažádá o překlad, pokud by server překlad neměl, odešle mi všechny možné překlady, které má a zároveň mi dá vědět, že nemá překlad, který by uživatel chtěl. Já poté uživateli nabídnu všechny jazyky, ve kterých může dostat přeložený text. Tento způsob sice bude přenášet větší množství dat, ale ušetří komunikaci mezi serverem a uživatelem, zároveň je tato možnost pro uživatele příjemnější než zobrazovat nativně třeba jen anglický jazyk.

2.7 Programovací jazyk

V zadání je, že výsledná aplikace by měla fungovat na platformě Android. Častými volbami pro Android aplikace jsou: Java (respektive Kotlin), Flutter nebo C#.

Kotlin je v dnešní době podle statistik nejrozšířenějším programovacím jazykem pro vývoj Android aplikací. Jedná se o nástavbu pro Javu, přímo určený pro vývoj mobilních aplikací.

Další možností je novější Flutter, jedná se o open-source SDK od společnosti Google, podporuje multiplatformní programování, to znamená, že jeden kód může fungovat jak na Android, tak na iOS.

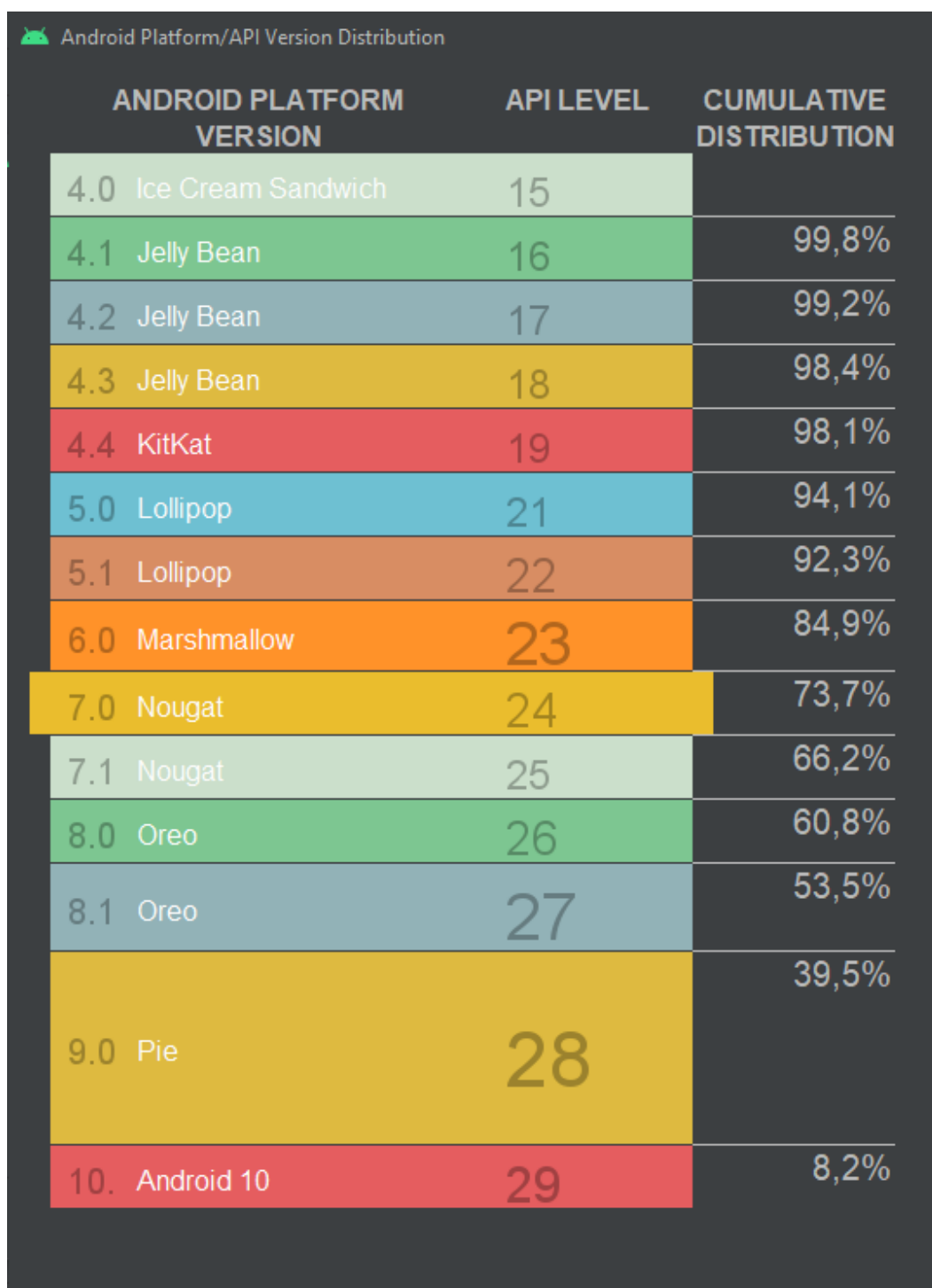
Poslední z vybraných jazyků je C#. Svoji strukturou a syntaxí je podobný Javě, proto se také využívá pro vývoj mobilních aplikací.

Vyzkoušel jsem si tutoriály pro mobilní aplikace ve všech třech jazycích a nejvíce mi ze všech vyhovoval Kotlin. Zároveň je Kotlin i nejrozšířenějším jazykem, proto věřím, že budou existovat již vyřešené problémy, na které narazím.

Odkazy:

- 1) <https://github.com/google-ar/arcore-android-sdk/issues/418>
- 2) <https://www.wikitude.com/store/>
- 3) <http://qrcodes.cz/qrkody-qr-code.php>
- 4) <https://github.com/dm77/barcodescanner>

- 5) <https://github.com/yuriy-budiyev/code-scanner>
- 6) <https://github.com/KingsMentor/MobileVisionBarcodeScanner>
- 7) [Distribuce android zařízení](#)
- 8) [https://upload.wikimedia.org/wikipedia/commons/0/08/Manhattan_distance.s
vg](https://upload.wikimedia.org/wikipedia/commons/0/08/Manhattan_distance.svg)
- 9) [http://web.archive.org/web/20170126150533/https://developers.google.com/
maps/articles/phpsqlsearch_v3](http://web.archive.org/web/20170126150533/https://developers.google.com/maps/articles/phpsqlsearch_v3)



////////////////////////////////////

Možnost přidat (nafotit) exponát a přidat do aplikace (uživatelé v aplikaci) Podporovat uživatele