

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

Bakalářská práce

Mobilní průvodce studenta na platformě iOS

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Stepan MOCJAK**
Osobní číslo: **A18B0274P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informatika**
Téma práce: **Mobilní průvodce studenta na platformě iOS**
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s možnostmi tvorby aplikací pro platformu iOS.
2. Vyberte vhodné funkcionality, které by měl mobilní průvodce studenta na ZČU poskytovat. Při výběru funkcionalit využijte mj. informační zdroje univerzity.
3. Navrhněte aplikaci mobilního průvodce studenta ZČU pro platformu iOS.
4. Navrženou aplikaci realizujte, ověřte její funkcionality a navrhněte možná rozšíření.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Ladislav Pešička**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **4. října 2021**
Termín odevzdání bakalářské práce: **5. května 2022**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 14. října 2021

Prohlášení

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

V Plzni dne 25. června 2022

Stepan Mocjak

Poděkování

Chtěl bych poděkovat svému vedoucímu bakalářské práce Ing. Ladislavu Pešíčkovi za odborné vedení, za pomoc a rady při zpracování této práce.

Abstract

Mobile student's guide for iOS. The goal of this thesis is to get acquainted with the creation of applications for the iOS platform, the selection of proper functionality, application design, and application implementation. The application should help students, teachers, and visitors to orientate in the UWB campus and simplify the process of finding rooms. A 3D campus model is modeled in Blender. Implementation was done in the game engine Godot and GDScript language.

Abstrakt

Cílem této práce je seznámení se s tvorbou aplikací pro platformu iOS, výběr vhodné funkcionality, návrh aplikace a realizace aplikace. Aplikace by měla pomáhat studentům, vyučujícím a návštěvám k orientaci v kampusu ZČU a ulehčit proces vyhledávání místností. Pro implementaci byl namodelován 3D model kampusu v aplikaci Blender. Práce byla realizována v herním enginu Godot a jazyku GDScript.

Obsah

1	Úvod	9
2	Výběr funkcionality	10
2.1	Návrh aplikace	10
2.2	Existující řešení	11
3	Vývoj mobilních aplikací	12
3.1	Android	13
3.2	iOS	13
3.3	Verze OS	14
4	Vývoj aplikací pro iOS	16
4.1	Nativní vývoj Xcode	16
4.2	Webový vývoj Ionic	16
4.3	Flutter	17
4.4	Unity vs Godot	17
4.4.1	Unity	18
4.4.2	Godot	18
4.5	Publikace aplikací iOS	18
5	Zdroje dat	20
5.1	API	20
5.1.1	Místnosti	20
5.1.2	Budovy	21
5.1.3	Rozvrh	21
5.2	Ostatní zdroje dat	22
6	Tvorba modelu	25
6.1	Blender	25
6.2	Modelování	25
7	Herní engine Godot	28
7.1	Využití	28
7.2	GScript	28
7.3	C#	29
7.4	Visual Scripting	30
7.5	Stavební prvky	30

7.6	Pluginy	32
7.7	Propojení s Xcode	33
7.8	Možnosti exportu v Godotu	33
8	Vývoj aplikace	35
8.1	Funkcionalita	35
8.1.1	Mapa	35
8.1.2	Vyhledání budovy a informace o místnosti	37
8.1.3	Informace o budově	40
8.1.4	Odkazy	41
8.2	Knihovny a zdroje	42
8.3	Animace	42
8.4	Databáze	42
8.4.1	Použité tabulky	43
8.5	Překlady	44
8.6	Aktualizace	44
8.7	Export/Instalace	45
8.7.1	Xcode	47
8.8	Nasazení	49
8.9	Rozšíření	50
9	Testování	51
9.1	Chování na různých zařízeních	51
9.2	Stabilita aplikace	51
9.3	První spuštění	51
9.4	Otočení obrazovky	51
9.5	Stíny v aplikaci	52
9.6	Vyhledání místnosti	53
9.7	Zobrazení místnosti	54
9.8	Stažení dat z API	54
9.9	Zobrazení budov	54
9.10	Odkazy	55
9.11	Změna jazyka	55
9.12	Shrnutí testování	55
10	Závěr	56
	Literatura	59

1 Úvod

Cílem práce je vytvořit mobilní aplikaci pro platformu iOS pro snadnou orientaci v kampusu ZČU. Jedná se o areál na Borských polích, kterému se jinak říká *Zelený trojúhelník*. Univerzita má celou řadu dalších budov i mimo kampus, ty ale aplikace nezobrazuje (informace o nich ale poskytuje). Celý areál je poměrně velký a pro nastupující studenty a návštěvy může orientace v něm působit potíže. Proto jsem pro lepší orientaci vybral pro zobrazení kampusu 3D model. Předpokládaná skupina uživatelů jsou lidé, kteří potřebují rychle najít umístění učebny podle jejího názvu (např. UC329). Mohli by toho využít například studenti, kteří přijedou na Erasmus. Existují aplikace pro zobrazení rozvrhové akce, bohužel ty pouze vypíšíou název místnosti. Uživatel tak musí vědět, kde se místnost nachází. Aplikace by měla uživatele o danou funkcionalitu obohatit.

Práce je zaměřená na možnosti vývoje aplikací pro platformu iOS. Zároveň vyberu funkcionalitu které by měla aplikace obsahovat a využiji informační zdroje univerzity. Platformu iOS jsem si vybral z důvodu potřeby pokrytí uživatelů mobilních zařízení od společnosti Apple. Pro konkurenční platformu Android vzniká analogická aplikace od kolegy Lukase Freye. Vyvíjet aplikace pro platformu iOS můžeme různými způsoby.

V práci projdeme porovnání a rozdíly v mobilních platformách. Porovnání způsobů vývoje pro platformu iOS a nástrojů které byly vybrány pro vývoj a proč. Jaké zdroje dat pro aplikace a univerzitní zdroje jsem našel a použil. V práci bylo popsáno, jak probíhalo modelování kampusu, vývoj aplikace a její testování s využitím testovacích scénářů.

2 Výběr funkcionality

Při zahájení studia na Západočeské univerzitě jsem měl problém s orientací v kampusu školy. Nebyl jsem sám, mnoho mých spolužáků také zažívalo tento problém. Častokrát se stalo, že jsem se pohyboval po fakultě strojní a hledal učebnu, která se reálně nacházela na fakultě aplikovaných věd. I nyní se mi to občas přihodí.

Pro sledování rozvrhu a zkoušek jsem využíval aplikaci *Student ZČU*. Ovšem v aplikaci mi chyběla možnost najít budovu, ve které se učebna nachází. Z mé zkušenosti a zkušenosti mých spolužáků vím, že tyto zkratky budov bylo těžké si zapamatovat. Proto jsem se rozhodl naprogramovat tuto aplikaci, která pomůže snadné orientaci v kampusu ZČU. Nemusí jí využívat pouze studenti, ale také například studenti z programu Erasmus, návštěvy (krátkodobé studijní a pracovní pobyty) nebo studenti středních škol při Dni otevřených dveří (DOD) nebo veletrhu pracovních příležitostí.

Aplikace by měla pokrývat uživatele zařízení od společnosti Apple s operačním systémem iOS.

2.1 Návrh aplikace

V aplikaci jsem se rozhodl reprezentovat mapu jako 3D model, který lze rotovat a naklánět ho. Výběr modelovacího softwaru a průběh modelování je popsán v kapitole 6.

Pro programování by mělo být vybráno vhodné vývojové prostředí pro systém iOS, které nabízí rozšířenou podporu pro práci s 3D modely. Porovnání možností a různých vývojových prostředí bude uvedeno v kapitole 4.

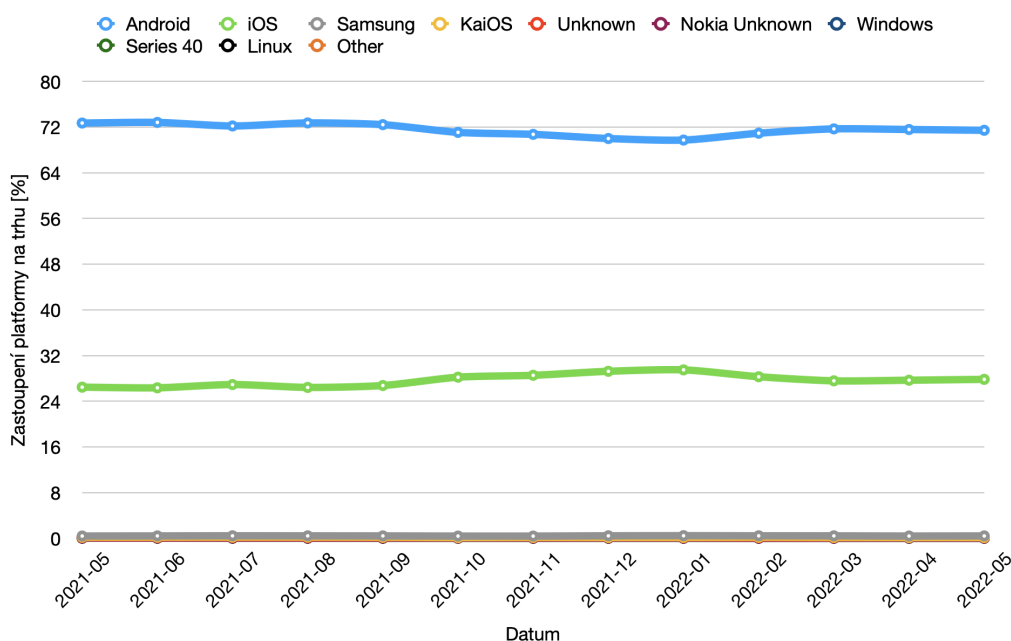
Aplikace by měla zobrazovat aktuální data místností. Proto by měla existovat možnost stáhnout data místnosti, tak aby byla aktuální (kapitola 5). Dále by neměl být problém aplikaci používat i bez připojení k internetu. K tomu by měla aplikace využívat interní uložení dat v podobě SQLite databáze (kapitola 8.4).

Aplikace by měla usnadnit hledání budovy. Po vyhledání místnosti by měla být možnost zobrazit budovu, ve které se místnost nachází. Budova by se měla na mapě (3D modelu) vizuálně zviditelnit. V detailu budovy by se měly zobrazovat fotografie budovy (pro jednodušší orientaci). Detaily ohledně implementace najdete v kapitole 8.1.3.

3 Vývoj mobilních aplikací

Mobilní aplikace lze vyvíjet v současné době pro dvě hlavní platformy: Android a iOS. Ostatní platformy jako například Windows Mobile, Symbian, BlackBerry OS jsou již nyní minulostí a procento využití je tak malé, že nemá cenu se jimi zabývat. Systém iOS je na druhé pozici v podílu zařízení na trhu. K březnu roku 2022 měl iOS podíl na trhu 27.57 % a Android 71.7 % (Obrázek 3.1). Aplikace pro iOS představují nezanedbatelnou část vývoje pro mobilní zařízení.

Aplikace lze vyvíjet různými způsoby v různých frameworkích a jazycích. Dokonce existují webové generátory aplikací, které vygenerují mobilní aplikaci například jenom z webového odkazu na stránky. Takovéto generátory pouze zabalí web z odkazu do aplikace. Často se využívají i šablony, které usnadní vývoj a vývojář pouze zajistí funkcionalitu (<https://ionicthemes.com>).



Obrázek 3.1: Graf podílu mobilních OS na trhu za poslední rok podle stránky statcounter.com

3.1 Android

Android je mobilní operační systém založený na jádře Linux. Je dostupný pod licenci Apache 2.0 (GPLv2 pro modifikace linuxového jádra). Na vývoji se z velké části podílí firma Google a směřuje jeho vývoj [4]. Výrobci mobilních telefonů si systém rádi upravují a přidávají k němu své nadstavby (MIUI, EMUI, LG UX, ...). Tyto nadstavby ale často systém zpomalují a výrobci pro koncová mobilní zařízení vytvářejí aktualizace příliš pomalu, nebo i v případě některých výrobců vůbec. Tím pádem se v populaci vyskytují zařízení s mnoha různými verzemi a vývoj pro systém Android je tak mnohem složitější. Je tady potřeba ošetřit nejenom mnoho různých rozlišení, velikostí obrazovek, ale i výjimky spojené s verzemi systému. Android se využívá nejenom na mobilních zařízeních, ale i v různých jiných zařízeních, jako je například televize, hodinky, automobil a další podobná technika [34].

Existují výrobci, jako Zebra, kteří si Android přizpůsobí do takové míry, že jej lze využívat v enterprise zařízeních. Takovýto Android je již ale výrazně upraven a například nedovoluje instalovat aplikace koncovému uživateli, nebo spouštět jiné aplikace, než které má povoleno [5].

Pro nativní vývoj aplikací pro Android se primárně používá Android Studio. Jedná se o vývojové prostředí postavené na populárním nástroji *IntelliJ IDEA* od společnosti JetBrains. V prosinci roku 2014 byla představena verze 1.0 a od té doby se jedná o nejpopulárnější nástroj pro vývoj Android aplikací.

Aplikace jsou primárně distribuovány přes Google Play. Nejedná se ale o jediný způsob distribuce. Oproti iOS se aplikace mohou stahovat rovnou z webu, nebo jiných zdrojů (příloha zprávy nebo e-mailu, odkaz ke stažení, a další).

3.2 iOS

Operační systém iOS je vyvíjen společností Apple. Systém lze využít na zařízeních od společnosti Apple, jako jsou iPhone, iPad a další mobilní zařízení. Vychází z operačního systému MacOS, který se vyskytuje na počítačích firmy Apple [28]. Systém je licencován pouze pro zařízení od společnosti Apple. Uzavřený systém se může zdát jako nevýhoda, ovšem společnost Apple se stará o vytváření a distribuci aktualizací systému pro svá zařízení po dlouhou dobu od jejího uvedení na trh. To napomáhá tomu, že se změny v OS rychle rozšíří a většina uživatelů má vždy nejnovější verzi. Díky tomu mají vývojáři méně práce s optimalizací svých aplikací na různé verze. Problematice verzí se věnuji v následující sekci.

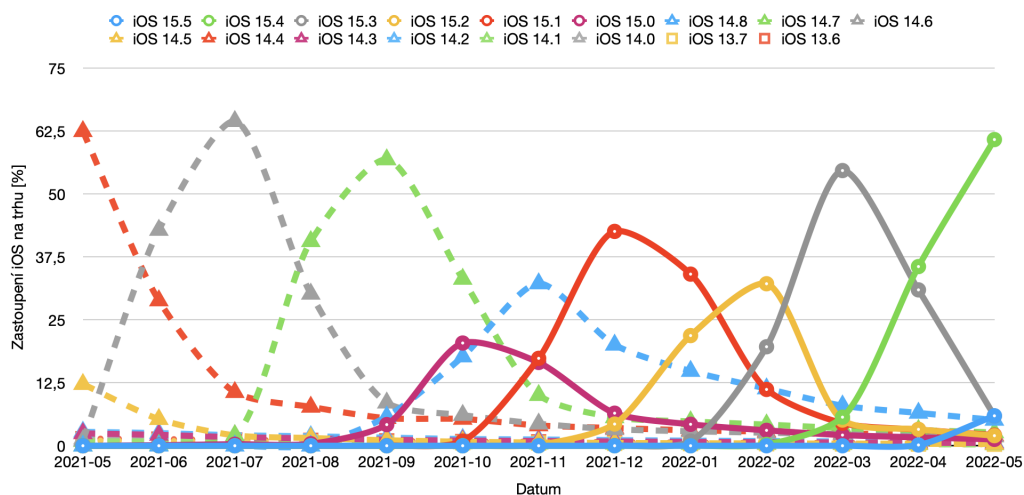
Pro vývoj aplikace je potřeba počítač s operačním systémem MacOS. Na vývoj se využívá nástroj Xcode, který se na MacOS dá stáhnout z App Store zdarma. Nativní vývoj v Xcode ovšem není jediná možnost. Existuje mnoho jiných frameworků ve kterých se vyvíjí. Ovšem vždy je nutnost mít nainstalovaný Xcode.

Aplikace jsou legální cestou distribuovány především přes App Store, ovšem existují i výjimky. Apple má také speciální program pro firmy, v němž dovoluji distribuci aplikací pouze pro určitá zařízení, například pro firemní telefony [2]. Aplikace je tak například distribuovaná pouze pro telefony vybrané společnosti a nemůže se dostat na veřejnost. Jedná se alternativu pro Android zařízení.

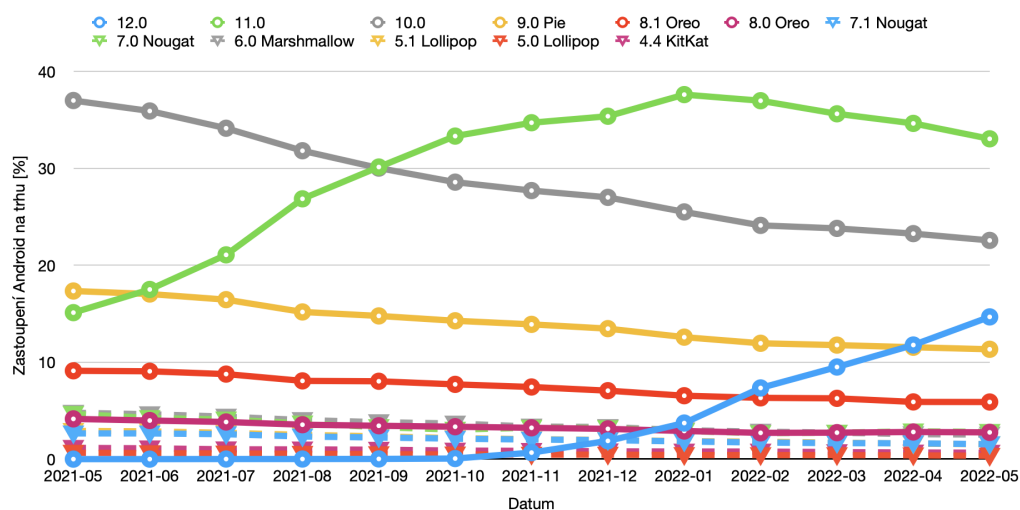
3.3 Verze OS

Největším problémem Androidu se zdá dlouhé a pomalé aktualizace systémů oproti iOS. Google se snaží výrobce telefonů podporovat, aby vyvíjeli aktualizace, a to díky různým rozdělením v systému. Specifické věci pro telefon (například ovladače kamery, sensorů, atd..) zůstávají během aktualizací neměnné. Takový program se zrodil například u Androidu 8.0 pod názvem *Project Treble* [38]. V teorii jsou takové věci dobré, v praxi bohužel výrobci vypustí telefon s jednou verzí Androidu a v rámci životnosti telefonu dostane zákazník jenom přibližně jednu aktualizaci systému navíc. Grafy zobrazené na obrázku 3.3 názorně ukazují rychlost nasazení nových verzí systémů v praxi. Nová verze Androidu má většinou půl roku zpoždění po vypuštění na to, aby se na trh vůbec dostala a další rok na to, aby se stala nejpoužívanější. V té době je již ale oznámená nová verze a začíná její distribuce. Vývojáři tak musí podporovat mnoho verzí Androidu nazpět a nemohou využívat naplno nové technologie, které jim přináší nové verze.

Pro iOS je to výrazně odlišné. Na obrázku 3.2 můžeme vidět jak se nové verze okamžitě distribují do koncových zařízení. Aktualizace dostávají i telefony starší pěti let [1]. To zajišťuje, že po celou dobu fyzické životnosti zařízení mají uživatelé nejaktuálnější iOS. Díky tomu mohou vývojáři okamžitě reagovat na novinky v aktualizaci. Také nemusí podporovat tolik předchozích verzí systému, jako v případě Androidu. Celý vývoj se díky tomu zjednodušuje a urychluje.



Obrázek 3.2: Graf podílu verzí iOS na trhu za poslední rok podle stránky statcounter.com.



Obrázek 3.3: Graf podílu verzí Android na trhu za poslední rok podle stránky statcounter.com.

4 Vývoj aplikací pro iOS

Vývoj pro iOS může probíhat buď nativně (Xcode) nebo multiplatformě (Flutter, Kotlin Multiplatform, Ionic, Unity, Godot a další). Ve všech případech je ale nutnost mít počítač od firmy Apple s MacOS a Xcode. Novinkou je nyní Swift Playgrounds. Dříve to byl nástroj pouze pro výuku jazyku Swift, nyní ovšem umožňuje jednoduchý vývoj aplikací na Apple iPad [42].

V této kapitole probereme různé možnosti vývoje pro platformu iOS. Při výběru frameworku je zohledněna rychlost vývoje, podporu, práci s 3D objekty pro zobrazení mapy a možnost rozšíření na další platformy.

4.1 Nativní vývoj | Xcode

Xcode lze stáhnout zdarma z Appstore. Aktuálně se jedná o verzi Xcode 13. Podporuje verzovací a teamové systémy jako Git nebo nově Xcode Cloud. Jedná se kompletní IDE s podporou pro jazyky C, C++, Objective-C a Swift [46]. Poslední jmenovaný je nejnovějším počinem společnosti Apple. Jedná se o objektově orientovaný jazyk, který má veliký potenciál a je určený především pro vývoj aplikací na iOS, iPadOS, MacOS, tvOS, a watchOS [41]. Xcode se pro tvorbu iOS aplikací jeví hlavním kandidátem. Ovšem pro vývoj 3D aplikací není příliš optimalizovaný [30]. Navíc pro možné budoucí rozšíření aplikací například na platformu Android je nevyhovující z důvodu využití svého specifického jazyka a vývojového prostředí.

Apple podporuje vývoj nativních aplikací a poskytuje k výuce oficiální návody [26]. Dále jsou k dispozici videolekce ze Stanfordské Univerzity [40] a Applem vyvinutá aplikace pro výuku jazyku Swift, která se jmenuje Swift Playgrounds [42].

4.2 Webový vývoj | Ionic

Ionic je webový multiplatformní framework. Pro frontend využívá HTML a CSS. Logika se programuje v JavaScriptu nebo TypeScriptu. Podporuje frameworky jako jsou Vue, Angular a React [25]. Aplikace funguje tak, že se v ní zobrazí webového okno a v něm naprogramovaná aplikace. Aplikace je oproti nativní aplikaci pomalejší a závislá na výchozím prohlížeči nastaveném v telefonu. Pro Android je to ve většině případů Chrome a pro iOS Safari a jádro Webkit [45]. Výhodou je ovšem rychlost vývoje. Vývojáři, kteří

vyvíjeli webové aplikace tak mohou rychle vytvořit i aplikaci mobilní.

Pomocí HTML5 lze zobrazovat 3D objekty, ovšem na mobilním zařízení je tento způsob zobrazování příliš pomalé a neintuitivní. Oproti nativnímu vývoji, zde není možné využít nativní API přes primární jazyk (u Ionicu se jedná o JavaScript) a přistupovat tak například k Bluetooth a podobně. Pro dané funkcionality se zde využívají naprogramované pluginy. Ovšem u nich se funkčnost může rozcházet v závislosti na platformě a toho, jak byl plugin naprogramován. Mnoho pluginů také nepodporuje všechny platformy a na různých platformách mohou vypadat jinak. Osobně se mi stalo, že například plugin ZBar na platformě iOS vypadal a choval se jinak, než na platformě Android. To bylo způsobeno tím, že se při akci spouštěla nativní aktivita s kamerou a na každé platformě vypadali nativní aktivity jinak.

4.3 Flutter

Flutter je multiplatformní framework vyvíjený společností Google sloužící pro tvorbu aplikací na platformy Android, iOS, web a mnoho dalších. Největší podporu mají Android a iOS, ovšem probíhá intenzivní vývoj frameworku pro web. Oproti Ionicu se aplikace programují v jazyku Dart, který se používá jak i pro frontend, tak i pro backend. Kód aplikace se kompiluje přímo pro ARM a Intel [15]. Výsledná aplikace je rychlejší, než v případě Ionicu [23]. Ovšem oproti nativnímu vývoji je aplikace stále trochu větší a o něco pomalejší.

Bohužel v této chvíli Flutter nepodporuje nativní zobrazování 3D objektů a neumí s nimi pracovat. Pro 3D objekty se v poslední době objevil plugin Flutter Cube [16]. Ovšem plugin není příliš rozšířený a chybí mu dokumentace. Proto i když s Flutterem mám několikaleté zkušenosti, tak jsem jej nezvolil v rámci mé práce.

4.4 Unity vs Godot

Z důvodu lepší podpory pro zobrazování 3D objektů bylo potřeba se podívat na pole herních enginů. Největší hráči na tomto poli jsou Unreal Engine, Unity a v neposlední řadě také Godot. Unreal Engine má nejlepší podporu funkcí pro 3D, bohužel pro 2D nemá tolik nástrojů [24]. A obzvláště kvůli jeho velikosti vývojového prostředí (15-50GB) byl zavrhnut. Rozhodovalo se tedy mezi herními enginy Unity a Godot.

4.4.1 Unity

Unity je 3D herní engine s možností exportu pro mnoho platforem. Unity je na trhu již od roku 2005, kdy vznikl jako herní engine určený pouze pro MacOS a od té doby nabral na popularitě mezi mnoho vývojáři. Nyní Unity podporuje 27 platforem, jako jsou Android, iOS, MacOS, Windows a další. Podpora 3D je na vysoké úrovni [27]. Unity je komerční program a přestože má verzi zdarma, je omezena mnoha podmínkami, mezi které patří například: výdělek aplikace do 100 000 \$ ročně, nemožnost odstranění loga Unity při načítání aplikace a další [43]. Pro další funkcionality je nutnost si zaplatit licence, které začínají na ceně od 399 \$ ročně. Dále Unity nemá podporu pro Git, pouze přes plugin [18].

4.4.2 Godot

Godot je OpenSource 3D herní engine. Podporuje širokou škálu programovacích jazyků jako třeba GDScript, C#, C++ [19]. Neoficiální podpora je například pro Python [37], Lua a mnoho dalších jazyků. Také má možnost tvořit pomocí vizuálních skriptů stejně jako třeba Unreal Engine. GDScript je jazyk vytvořený pro potřeby Godotu a je plně podporovaný. Je to skriptovací jazyk podobný Pythonu a také z něj vychází [17]. Godot je dostupný pod MIT licencí, což znamená, že je možné ho použít pro jakýkoli účel, i komerční. Oproti Unity to znamená volné použití bez nutnosti mít předdefinovanou obrazovku při načítání aplikace a není zde limit na to, kolik peněz mohu s aplikací vydělat [20].

4.5 Publikace aplikací iOS

Aplikace pro iOS se oficiálně publikují přes Apple App Store [29]. Pro publikování v tomto obchodě je vyžadováno, aby vývojářský účet (Apple ID) byl součástí Apple Developer Programu. Členství v programu stojí 99 \$ ročně a vyžaduje aby jste byli starší 18 let [8]. Pro neziskové společnosti, školy a státní správu mohou být poplatky prominuty. Musí se ale žádat o prominutí a čekat na ověření a schválení [3].

Před vlastní publikací musí aplikace splňovat některá pravidla, aby mohla být vůbec schválena. Některá z nich jsou vypsána níže [29]:

1. Bezpečnost - Aplikace musí být pro užívání bezpečná a pro uživateli nesmí způsobit ani fyzickou nebo psychickou újmu. Aplikace by neměla obsahovat diskriminační obsah, či jakkoliv urážlivý obsah (nábožen-

ství, orientace, pohlaví, ...). Zakázaný je také jakýkoli pornografický nebo sexuální materiál.

2. Fyzické poškození - Aplikace nesmí způsobit uživateli žádnou fyzickou újmu. Aplikace tak nesmí lživě uvádět, že funguje jako rentgen, měří krevní tlak, nebo hodnotu glukózy v krvi, pokud používá pouze senzory v telefonu. Také nejsou povoleny aplikace podporující užívání tabákových nebo alkoholických výrobků.
3. Funkcionalita - Aplikace musí být dodaná s plně funkčními URL odkazy. Neměla by obsahovat škodlivý kód nebo viry. Nesmí spouštět kód přidaný odjinud nebo stažený z internetu. Pokud aplikace simuluje domovskou obrazovku nebo widgety, bude také zamítnuta. Pokud obsahuje reklamní banery, pro proces schválení by měly být plně funkční a neobsahovat pouze prázdná pole nebo testovací reklamy.

Na všechny tyto body bylo během návrhu a vývoje aplikace myšleno. Aplikace by tak měla splňovat všechny podmínky pro publikaci na App Store. Detailnější popis publikace aplikace a jaká úskalí se tam nachází lze nalézt ve knize *Vývoj aplikací pro iOS* od Luboslava Lacka [29].

5 Zdroje dat

Pro vývoj aplikace bylo žádoucí využít různě zdroje dat, ať již počítám podklady pro mapy, nebo univerzitní zdroje (učebny, rozvrhy, budovy, ...). Většina potřebných dat ke zpracování aplikace jsou volně k dispozici na internetu. Jedná se například o mapové podklady, ze kterých byl vytvořen model nebo fotografie kampusu s popsányými budovami (obrázek 2.1). Ovšem některé specifické zdroje pro ZČU jsou dostupné pouze přes API.

5.1 API

Aplikace by měla primárně zobrazovat budovy a informace o učebně. Proto byli využity univerzitní zdroje a to konkrétně IS/STAG. Dokumentace k němu se také nachází na portálu IS/STAG. Primárně jsem se zaměřil na získávání informací ohledně místností a informací ohledně budov. Pro testování tohoto API byl využit program Postman, který je v základní verzi zdarma k použití.

Na této adrese lze najít demo verzi API která je určena pro testování:

- <https://stag-demo.zcu.cz>

Daná adresa byla využita pro vývoj a testování.

Pro produkční verzi byla využita tato adresa:

- <https://stag-ws.zcu.cz>

Tato adresa se využívá v produkční verzi celé aplikace.

Na adrese níže se nachází dokumentace k API IS/STAG. Jednalo se o můj primární zdroj informací k tomuto API a jak s ním pracovat.

- <https://stag-ws.zcu.cz/ws/web>

5.1.1 Místnosti

- `/ws/services/rest2/mistnost/getMistnostiInfo`

Tento endpoint je využitelný pro získání informací ohledně místností. Konkrétně vrací například tyto informace:

- `zkrBudovy` - Zkratka budovy (např. UL)

- `cisloMistnosti` - Číslo místnosti (např. 409)
- `katedra` - Zkratka katedry (např. SKM)
- `pracoviste` - Pracoviště (např. Koleje a menzy)
- `kapacita` - Kapacita (např. 50)
- `typ` - Typ místnosti (např. Kancelář)
- `plocha` - Plocha místnosti (např. 17.73)
- `podlazi` - Číslo podlaží (např. 2)
- `spolecnyFond` - Zda je místnost ve společném fondu

Výše vypsané informace nejsou jediné, které endpoint vrací. Jedná se pouze o výtah informací použitých v aplikaci.

5.1.2 Budovy

- `/ws/services/rest2/ciselniky/getCiselnik?domena=BUDOVA`

Pro získávání informací ohledně budov neexistuje podobný endpoint, jako pro získávání místností. Proto jsem využil tento obecný endpoint s parametrem *domena* a jeho hodnotou *BUDOVA*.

- `ciselnikItem` - seznam všech budov
 - `domena` - Název hledané domény (BUDOVA)
 - `zkratka` - Vždy má hodnotu *null*
 - `nazev` - Název budovy, většinou adresa budovy (např. Husova 11)
 - `key` - Dvoupísmenná zkratka budovy (např. HJ)
 - `poradi` - Vždy s hodnotou *999*

5.1.3 Rozvrh

- `/ws/services/rest2/rozvrhy/getRozvrhByMistnost?budova=EP&mistnost=130`

Rozvrhy pro jednotlivé místnosti lze jednoduše získat díky tomuto endpointu. Musíme ho ovšem volat s parametry *budova* a *mistnost* kam vložíme zkratku budovy a číslo místnosti.

- rozvrhovaAkce - seznam všech rozvrhových akcí pro danou místnost
 - nazev - Název rozvrhové akce (např. Nultý týden FAV, doc. Váša)
 - katedra - Název katedry (např. KEP)
 - predmět - Název předmětu (např. TEL1)
 - typAkce - Typ akce (např. Přednáška)
 - typAkceZkr - Zkratka akce (např. Př)
 - semestr - O jaký semestr se jedná (např. ZS)
 - den - Název dne ve který se akce koná (např. Pondělí)
 - denZkr - Zkratka dne (např. Po)
 - datum, datumOd, datumDo - Datumy, kdy se akce koná (např. "value": "20.9.2021")

Nejedná se o výpis všech informací, které endpoint vrací. Aplikaci lze dále rozšířit o mnoho dalších zajímavých informací, které endpoint vrací. Například o jméno učitele, hodinu, vyučovací jazyk a mnoho dalších.

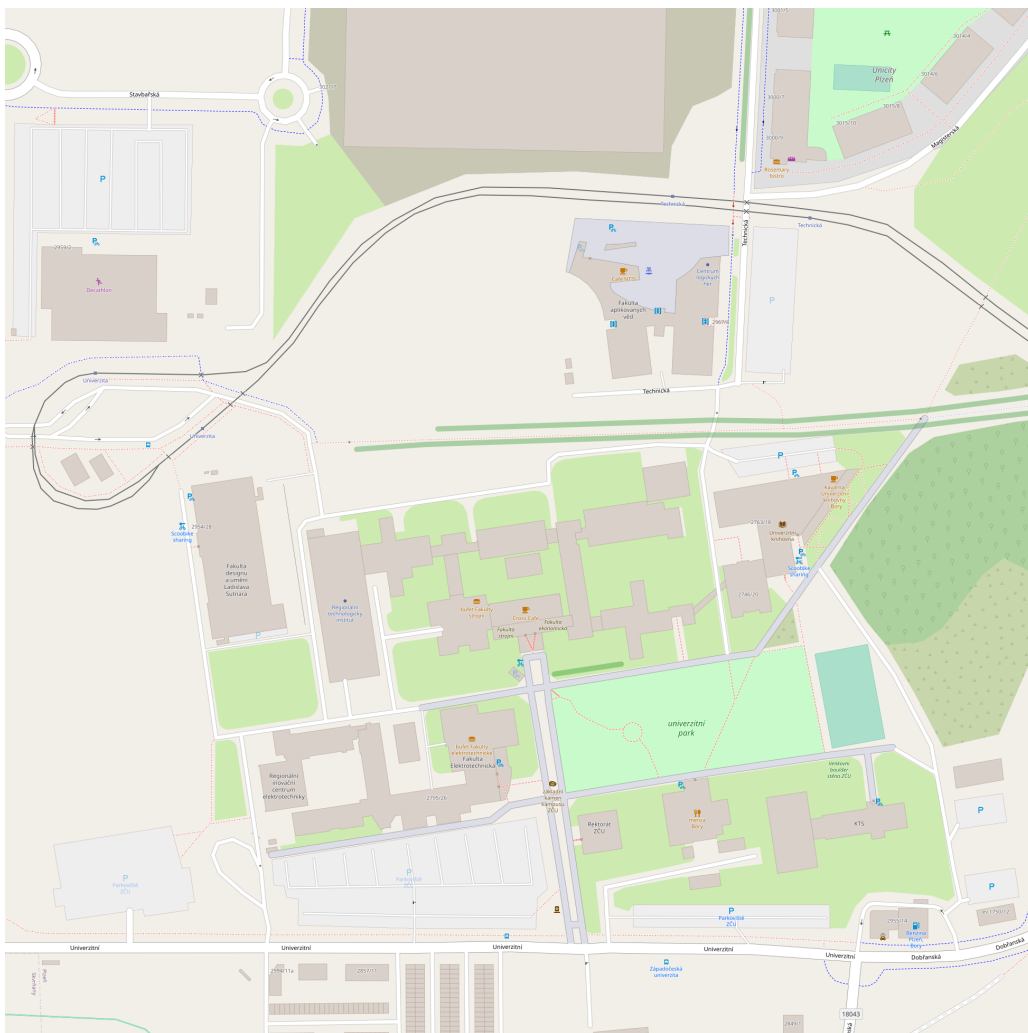
Také byl využit endpoint pro generování PDF s rozvrhovými akcemi pro danou učebnu. Ten se nachází na této adrese:

- [/ws/services/rest2/rozvrhy/getRozvrhByMistnostPDF](#)

U učebny s rozvrhovými akcemi se zobrazí tlačítko, které po rozkliknutí otevře tuto adresu ve výchozím prohlížeči systému iOS. Standardně se v novém okně prohlížeče zobrazí PDF, které můžete prohlížet. Některé prohlížeče se mohou pokoušet PDF stáhnout.

5.2 Ostatní zdroje dat

Další zdroje, které byli využité pro tvorbu aplikace byla například mapa, kterou jsem využil pro modelování kampusu. Mapa byla získána z otevřeného mapového zdroje - Open Street Map (obrázek 5.1).



Obrázek 5.1: Mapa, podle které se modelovaly budovy v aplikaci Blender.
Zdroj: <https://www.openstreetmap.org/#map=16/49.7377/13.3575>.

Detaily ohledně budov a místností byly získány z univerzitních webů. Zde byly nalezena například mapa budov s jejich pojmenováním (obrázek 2.1). Dalšími informacemi byly například otevírací doba, telefon na vrátnici a podobně.

Aplikace také zobrazuje fotografie některých budov. To napomůže uživateli lépe se zorientovat v reálném prostředí. Tyto fotografie jsem sám nafotil na telefon iPhone 11. Příklady fotografií použitých v aplikaci najdete na obrázku 5.2 a 5.3.



Obrázek 5.2: Fotografie budovy FAV použítá v aplikaci pro lepší orientaci v prostoru.



Obrázek 5.3: Fotografie budovy menzy.

6 Tvorba modelu

Pro intuitivní zobrazení kampusu, zasazení budovy do kontextu celého areálu, jsem zvolil 3D zobrazení. Proto jsem vytvořil 3D model. 3D model je matematická reprezentace nějakého třídímního objektu. Tyto 3D modely se využívají například pro reprezentaci reálného světa na PC, různých uměleckých děl a simulací nebo videohrách a mnoha dalších sférách [10].

Pro tvorbu 3D modelů existuje mnoho modelovacích softwarů. Modely se také můžou vytvářet pomocí fotogrammetrie. Jedná se o vědu, která se zabývá měřením pomocí fotografií [44]. Tímto způsobem se například vytvářejí modely budov pro Google Earth [32].

Výčet některých softwarů pro tvorbu 3D modelů najdete níže:

- Blender
- Cinema 4D
- Autodesk Maya
- Autodesk 3DS Max
- Wings 3D

Z výše popsaných mám zkušenosti pouze s Blenderem a Wings 3D. Jelikož mi práce s Wings 3D přišla neintuitivní a neexistuje k němu příliš návodů, využil jsem modelovací software Blender.

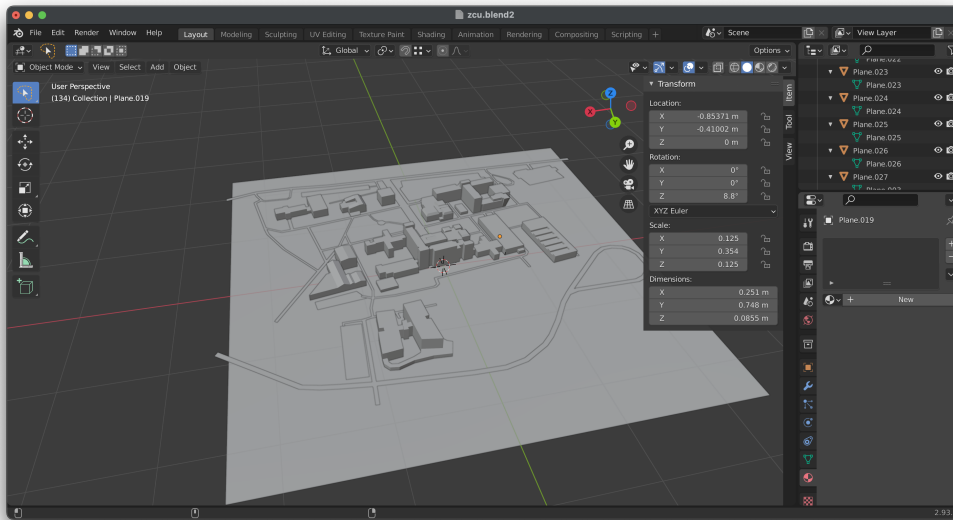
6.1 Blender

Ze všech uvedených softwarů mi přišel nejvhodnější Blender a to z podobných důvodů jako Godot. Blender je open source projekt, který má za sebou mnoho let vývoje. Je zdarma ke stažení na jejich oficiálních stránkách. Je vydáván pod MIT licencí [6]. Podporuje mnoho operačních systémů, jako MacOS, Linux, Windows a další [22]. V nové verzi má navíc vylepšené ovládání, které je mnohem intuitivnější, než původní. Ukázkou toho, jak aplikace vypadá můžete vidět na obrázku 6.1.

6.2 Modelování

Pro modelování byla využita podkladová mapa z Open Street Maps (obrázek 5.1). Následně jsem začal modelovat každou budovu zvlášť tak, že jsem její

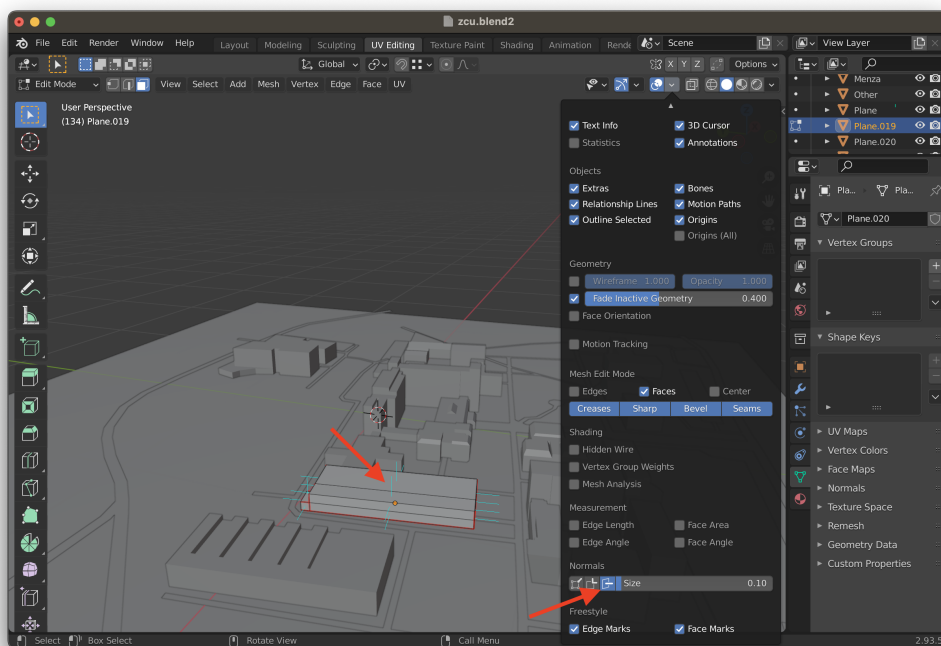
obrys obkreslil z mapy. Pro složité tvary budov, jako se vyskytují u budovy RICE, kde je budova speciálně nakloněna, jsem využil jako referenci Google Maps a jejich 3D pohled na mapu [32]. Při modelování jsem každou budovu (část budovy) oddělil. Toho jsem následně využil v aplikaci, kdy se budova "rozblíká". Cesty, tramvajové koleje a silnice dokreslil kolega Lukas Frey. Ukázku modelu v prostředí Blender můžete vidět na obrázku 6.1.



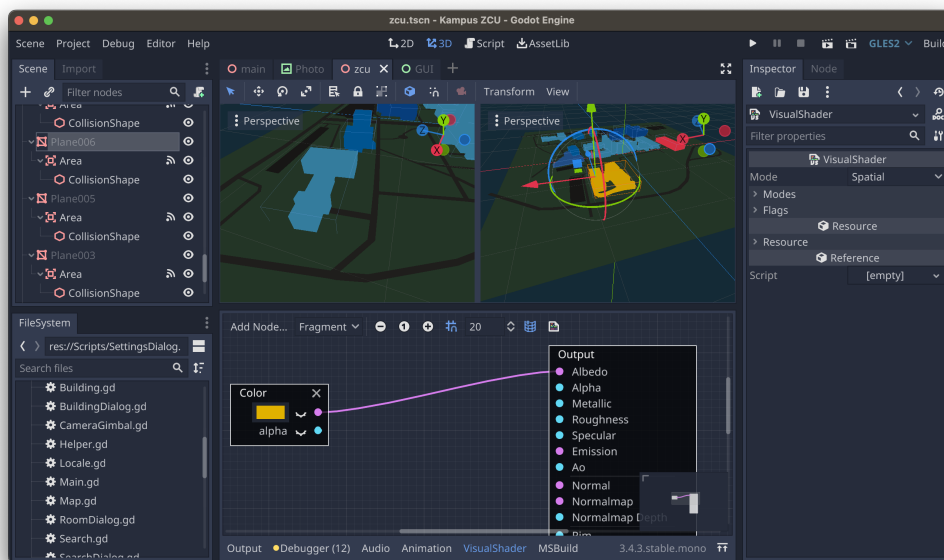
Obrázek 6.1: Snímek okna Blenderu s modelem kampusu ZČU.

Při tvorbě modelu jsem se setkal s několika problémy. Jedním z nich byli například normály u stěn. Některé stěny byly otočené do vnitřku modelu. Problém byl v tom, že v Blenderu jsem tyto normály při tvorbě neviděl. Důsledkem otočených normál byly artefakty v aplikaci. To jsem ovšem nevěděl a až po nějaké chvíli jsem zjistil, že se jedná o problém s normály. V Blenderu jsem musel následně povolit zobrazování normál (obrázek 6.2) a u špatně otočených ploch je ručně otočit.

Obarvení budov a ostatních objektů jsem provedl tak, že jsem jim přidal materiály. Materiály udávají objektu jeho vzhled [22]. Materiály jsem ovšem nepřidával v Blenderu, i když je to možné, ale v Godotu, kde jsem okamžitě viděl, jak bude celá scéna vypadat. Ukázku vytváření materiálu a jednoduchého shaderu pomocí *Visual Shader* editoru v Godotu můžete vidět na tomto obrázku 6.3. Barvy budov jsem vybral z fakultních webů.



Obrázek 6.2: Blender s povoleným zobrazováním normál u ploch.



Obrázek 6.3: Editace shaderu materiálu pomocí *Visual Shader* editoru Godotu.

7 Herní engine Godot

Pro vývoj byl využit herní engine Godot a to především z důvodů popsaných v předchozích kapitolách. Jedná se open source projekt pod licencí MIT [20]. Velikost, kterou zabírá na disku, je oproti jiným herním enginům, malá (okolo 30MB) [19]. Podporuje vývoj pro systému iOS a jednoduše se v něm pracuje s 3D objekty.

To byl jeden z hlavních důvodů, proč jsem si jej vybral. Jelikož je určen primárně pro 3D vývoj, existuje pro něj mnoho návodů. Dokumentace k němu je obsáhlá a obsahuje téměř vše, co jsem jako vývojář potřeboval.

7.1 Využití

Godot se především používá pro tvorbu her, ale není to jediné využití. Lze ho také využít na tvorbu 3D aplikací, v architektuře, ve zdravotnictví nebo simulaci různorodých situací. Existuje například aplikace určená pro kreslení pixelových obrázků [36]. Jelikož je projekt Open Source, je dostupný zdarma a vyvíjí ho lidé z celého světa [20].

Na projekt přispívají různé společnosti. Jednou z nich je například společnost Meta, která v něm vidí potenciál na poli virtuální a rozšířené reality. Další společností, která zainvestovala do Godotu, je například Microsoft. Díky jejich příspěvku se do enginu přidal programovací jazyk C# [11]. Díky tomu se Godot stává přímou alternativou hernímu enginu Unity, který C# využívá.

7.2 GDScript

Jedná se o nejpoužívanější jazyk v tomto enginu. Je to scriptovací jazyk, který je vyvinut pro potřeby vývoje v Godotu. Svoji syntaxí se velice podobá jazyku Python [17]. Godot má pro něj v sobě také přímo integrovaný editor a debugger. To z něj dělá univerzální "*All In One*" řešení pro tvorbu her a aplikací. Tento jazyk jsem využil i já v rámci tvorby aplikací.

Níže například můžete vidět ukázkou kódu definování proměnných 7.1, nebo funkcí 7.2.

```
1   var a = 5
2   var s = "Hello"
3   var arr = [1, 2, 3]
4   var dict = {"key": "value", 2: 3}
5   var typed_var: int
6   var inferred_type := "String"
```

Kód 7.1: Definování proměnných v GDScriptu.

```
1   func some_function(param1, param2):
2       var local_var = 5
3
4       if param1 < local_var:
5           print(param1)
6       elif param2 > 5:
7           print(param2)
8       else:
9           print("Fail!")
10
11      for i in range(20):
12          print(i)
13
14      while param2 != 0:
15          param2 -= 1
16
17      var local_var2 = param1 + 3
18      return local_var2
```

Kód 7.2: Definování funkce v GDScriptu.

Ukázky jazyku GDScript jsou převzatá z oficiální dokumentace [17].

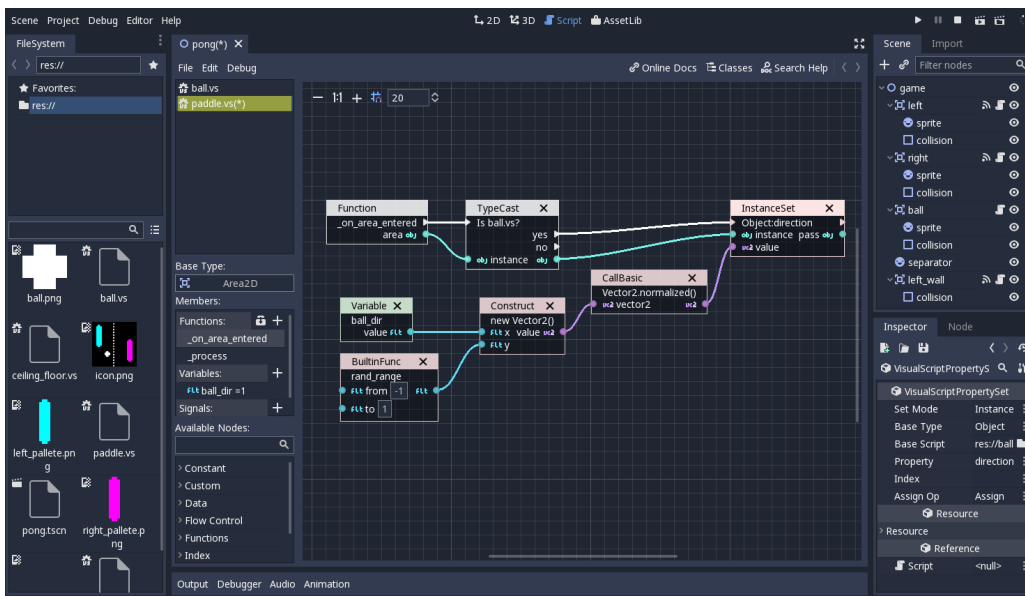
7.3 C#

Druhý v Godotu aktivně používaný jazyk je C#. Objevil se od verze 3.0, tudíž se jedná o novinku a můžou se zde vyskytnou některé chyby. Jazyk je implementován s .NET frameworkem Mono 6.x a podporuje C# ve verzi 8.0 [7]. Pro správnou funkci s podporou C# se ovšem musí z oficiální stránky

stáhnout Godot "Mono version". Ten podporuje jak C#, tak i ostatní jazyky jako GDScript. Tato verze je ovšem o něco větší než verze bez C#.

7.4 Visual Scripting

Další, méně používaný způsob psaní logiky nabízí Visual Scripting. Jedná se o ideální řešení pro tvůrce her, kteří se teprve seznamují s programováním nebo programovat neumí. Jedná se o jednoduchý vizuální editor, kde "přesouváte" různé komponenty jako jsou například proměnné, funkce, signály a podobně. Všechny komponenty potom podle potřeby pospojíte a z toho vám vzniká funkčnost. Ukázkou toho, jak vypadá takové scriptování můžete vidět na obrázku 7.1. Tyto skripty jsou kompatibilní se všemi ostatními jazyky Godotu, tudíž je lze využít například pro jednoduché funkčnosti a na složitější potom využít GDScript, nebo C#.

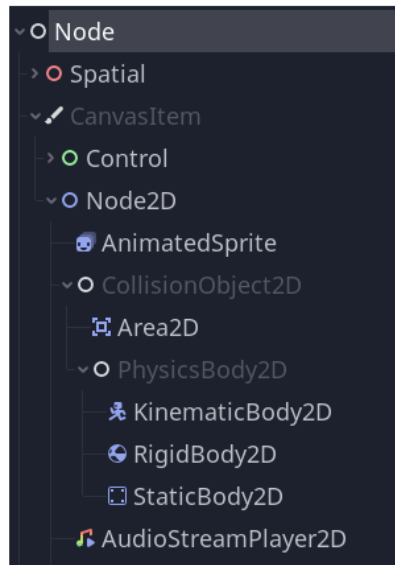


Obrázek 7.1: Ukázka editoru a vizuálního scriptování. Zdroj: <https://docs.godotengine.org/>.

7.5 Stavební prvky

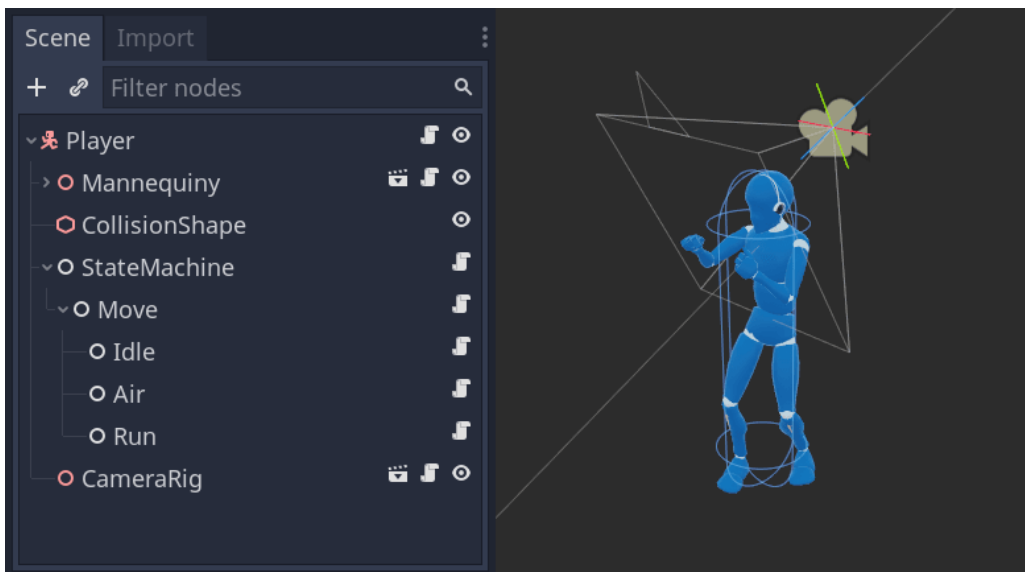
Základním stavebním prvkem Godotu je Node. Reprezentuje určitou funkčnost, jako je například zobrazení 3D objektu, animace, GUI a pod. Každý Node obsahuje vlastnosti, které určují chování objektu. Jedná se například o

polohu, otočení, zarovnání, nebo jiné, specifické. Nody se dají do sebe vnořovat a také dokáží dědit vlastnosti. Ke každému Nodu se dají přidat Scripty, které vlastnosti Nodu dokáží rozšířit [33]. Ukázku různých Nodů naleznete na obrázku 7.2.



Obrázek 7.2: Ukázka Nodů s různou funkcionalitou. Zdroj: <https://docs.godotengine.org/>.

Uskupení Nodů ve stromové struktuře sjednocujeme pod Scénou. Tu lze uložit a její instanci lze využít i několikrát v různých částech aplikace [33]. V každé aplikaci je minimálně jedna Scéna, ale v praxi se jich vyskytuje hned několik. Scény se ukládají jako soubory. Pod jednou scénou lze například ve hře uložit postavu hráče. Ukázku scény můžete vidět na obrázku 7.3.



Obrázek 7.3: Ukázka Scény hráče, která obsahuje různé Nody. Zdroj: <https://docs.godotengine.org/>.

7.6 Pluginy

Godot podporuje pluginy. Jedná se o oddělené funkční celky, které rozšiřují editor o nějakou funkčnost, nebo o nativní podporu. Jeden z takových je například plugin pro podporu verzovacího systému GIT.

Existují ovšem různé jiné pluginy a také pluginy specifické pro exportované platformy. Pro platformu iOS již existují mnoho pluginů, které podporují:

- Push Notifikace
- ARkit
- Kameru
- Game Center
- iCloud
- inAppStore
- PhotoPicker

Pro vývoj dalších pluginů má Godot připraven template. Sám jsem si možnost tvorby vyzkoušel a vytvořil jednoduchý plugin, který ze Swiftu

přepošle proměnnou do GDScriptu. V dalším vývoji bych rád rozšířil aplikaci o podporu GPS, pro kterou zatím neexistuje Plugin.

7.7 Propojení s Xcode

Pro testování aplikace na reálném zařízení jsem zpočátku exportoval celý projekt, ten následně otevřel v Xcode a až potom spustil v telefonu. Tento postup byl ovšem velice časově náročný. Pro každou změnu, kterou jsem provedl a chtěl otestovat, bylo potřeba tento postup stále opakovat.

Naštěstí tvůrci Godotu na tento problém mysleli. V dokumentaci jsem našel návod na propojení celého Godot projektu s Xcode. Díky tomuto propojení jsem aplikaci mohl spouštět na reálném zařízení mnohem rychleji a testovat mnohem častěji. Vývoj se díky tomu zrychlil. Detailní návod na propojení najdete na této adrese: https://docs.godotengine.org/en/stable/tutorials/export/exporting_for_ios.html.

Bohužel jsem posléze zjistil, že během tohoto propojení neotestuji reálné chování, pokud budu chtít aplikaci exportovat. Popis chyb a jejich opravu jsem popsal níže v kapitole 8.7.

7.8 Možnosti exportu v Godotu

Exportovat projekt z Godotu lze pro různé platformy. Pro export jsou vyžadované šablony. Šablony jsou optimalizované verze enginu bez editoru a předkompilované pro specifickou platformu [12]. Podporované platformy jsou aktuálně:

- Linux/MacOS/Windows
- UWP
- Android
- iOS
- HTML5

Oficiální podpora pro export na konzole se kvůli licencím na Godot nedostane. Ovšem pro XBox jde vyexportovat UWP aplikace a například pro Nintendo Switch a Playstation jde vyexportovat HTML5 verze a lehce upravit. Případně pro nativní podporu konzolí existují společnosti, které se zabývají tím, že projekt portují a vydávají na těchto platformách [9].

Godot standardně exportuje svůj hlavní balíček jako *.pck soubor, ve kterém jsou uloženy všechny scripty, scény, modely a podobně. Tyto soubory lze exportovat samostatně a díky tomu lze pro aplikace vytvářet módy, patche a podobně [13]. Pro *.pck soubory existuje alternativa v podobě ZIP archívů. U nich je výhoda menší velikosti díky kompresi, ale při používání aplikace mají problém s malou rychlostí čtení a zápisu [14]. Díky tomu se aplikace může snadno zaseknout.

8 Vývoj aplikace

Zpočátku jsem se aplikaci pokoušel naimplementovat nativně, ve vývojovém prostředí Xcode. To mi ovšem nevyhovovalo a pro vývoj 3D aplikace ani neexistovalo mnoho dokumentace a příkladů. Proto jsem se následně zaměřil na hledání nástroje, který mi práci s 3D modelem usnadní.

Implementace aplikace tedy proběhla v herním engineu Godot. I když se jedná především o herní engine, praxe ukázala, že je reálně použitelný i pro vývoj aplikací. Obzvláště takových, kde se využívají 3D modely. Práce s modely na takovéto platformě je mnohem přívětivější díky nástrojům, které přináší. A z těchto důvodů jsem si ho vybral pro implementaci aplikace. Většinu aplikace jsem napsal v jazyku GDScript. Dále jsem využil jazyk SQL pro práci s databází.

8.1 Funkcionalita

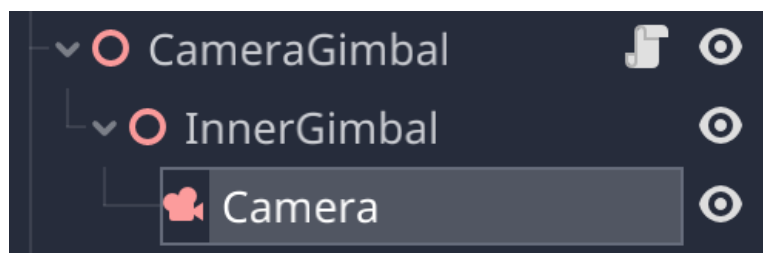
V této kapitole je popsán výběr a implementace funkcionality, kterou aplikace poskytuje.

8.1.1 Mapa

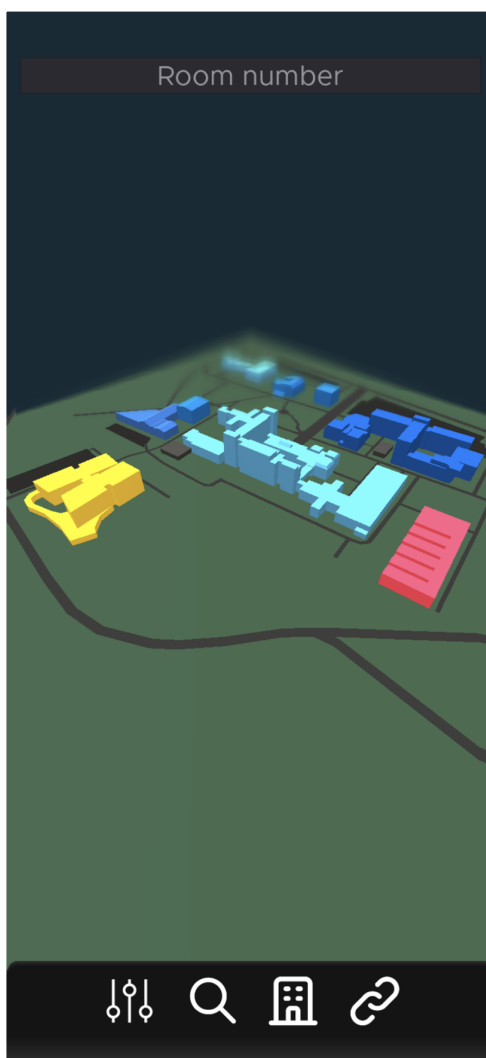
Mezi hlavní priority patřila jednoduchost a srozumitelnost zobrazení kampusu. Proto jsem pro zobrazování mapy vybral 3D model a ne 2D mapu nebo pouze textový popis. Po zapnutí aplikace se uživateli zobrazí mapa (3D model) se kterou může otáčet, nebo klikat na budovy (obrázek 8.2). Pro modelování jsem využil modelovací software Blender.

Uživatel může s mapou otáčet do stran a částečně i nahoru a dolů. Toho bylo docíleno pomocí dvou v sobě vnořených "*gimbalů*" (obrázek 8.1) a skriptu (kód 8.1), který obstarává uživatelské *inputy* v podobě doteků. Pojem gimbal zde používám jako bod, který je určen pro rotační pohyb v jedné ose. Optimalizace skriptu zabralo mnoho času a při jeho programování jsem přišel na chybu v Godotu. Pokud je počáteční úhel Y vnitřního gimbalu nastaven na jinou hodnotu, než je 0 a uživatel intenzivně krouží prstem po displeji, začne Godot vracet nesmyslné hodnoty pro úhel Y . Zdroj této chyby se mi zatím nepodařilo objevit, po dlouhém zkoumání jsem ale přišel na to, že úhel Y musí být nastaven na 0 a počáteční úhel jsem nastavil pomocí vnějšího gimbalu. Chybu jsem zatím nereportoval, v rámci dalšího vývoje

zainvestují do analýzy chyby více času a následně jí reportují jak na oficiálním Discord serveru, tak v repozitáři Godotu, který se nachází na GitHub.



Obrázek 8.1: Dva gimbaly a kamera. K prvnímu je přiřazen script, který se stará o zpracování uživatelských inputů a rotaci kamery.



Obrázek 8.2: Aplikace po zapnutí. Zobrazení 3D mapy (modelu).

```

1   func _unhandled_input(event):
2       var x_rotation = 0
3       var y_rotation = 0
4       if event is InputEventScreenDrag:
5           x_rotation -= (event.relative.x)/4
6           y_rotation -= -(event.relative.y)/4
7
8       rotate_object_local(Vector3.UP, x_rotation * rotation_speed
↪ * delta)
9       if ($InnerGimbal.rotation_degrees.x >= y_min_rot or
↪ y_rotation > 0) and ($InnerGimbal.rotation_degrees.x <= y_max_rot or
↪ y_rotation < 0):
10          $InnerGimbal.rotate_object_local(Vector3.RIGHT,
↪ y_rotation * rotation_speed * delta)
11
12          if $InnerGimbal.rotation_degrees.x < y_min_rot:
13              $InnerGimbal.rotation_degrees.x = y_min_rot
14          if ($InnerGimbal.rotation_degrees.x > y_max_rot):
15              $InnerGimbal.rotation_degrees.x = y_max_rot

```

Kód 8.1: Ukázka funkce, která se stará o rotaci s kamerou. Napsáno v jazyku GDScript.

8.1.2 Vyhledání budovy a informace o místnosti

Hlavní prioritou u aplikace pro mě byla jednoduchost toho, jak rychle může uživatel najít budovu jenom podle názvu. Toho docílí pouze zadáním budovy do hledacího pole, rozkliknutím místnosti a kliknutím na tlačítko *Zobrazit budovu*. Následně se uživateli budova rozbliká.

Po rozkliknutí místnosti se zobrazí okno *Detail místnosti*, kde naleznete informace o místnosti (obrázek 8.3). Jedná se například o označení budovy, číslo dveří, katedry a podobně. Dále se uživateli zobrazí jednoduchý rozvrh místnosti. Zobrazují se pouze akce, které se konají buď dnes, nebo v budoucnu. Pokud uživatel chce, může si celoroční rozvrh zobrazit ve formátu PDF, přes tlačítko *Otevřít PDF*. Po jeho stisknutí se otevře výchozí prohlížeč systému iOS s rozvrhem ve formátu PDF.

Níže je také ukázka funkce, starající se o našeptávání a zobrazování místností (kód 8.2).



Obrázek 8.3: Okno s detailem místnosti. Obsahuje informace o místnosti a rozvrh místnosti.

```

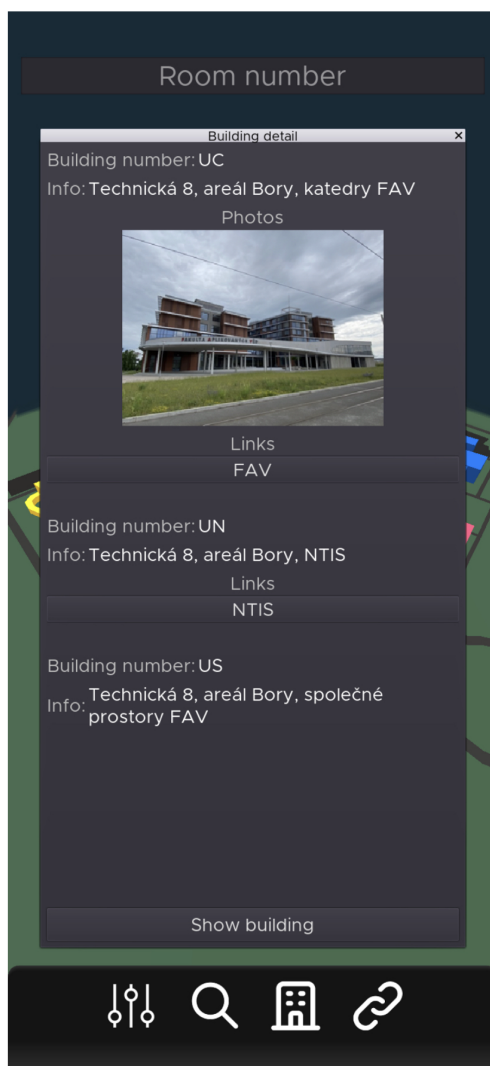
1  #Funkce volající se při změně textu v editačním panelu
2  func _on_LineEdit_text_changed(new_text):
3      if search_item == null : return
4
5      #Smazání listu výsledků
6      for n in $VBoxContainer/ScrollContainer/Items.get_children():
7          $VBoxContainer/ScrollContainer/Items.remove_child(n)
8          n.queue_free()
9
10     if new_text.empty() : return
11
12     var rooms = Database.search_rooms(new_text)
13
14     #Přidání tlačítek místností do seznamu pod editačním panelem
15     for room in rooms :
16         var new_item = search_item.duplicate()
17         new_item.visible = true
18         new_item.text = room["Name"]
19         new_item.connect("pressed", self, "_on_SearchItem_pressed",
↪ [new_item, room])
20         $VBoxContainer/ScrollContainer/Items.add_child(new_item,
↪ true)
21
22     pass

```

Kód 8.2: Funkce hledající místnosti při zadávání textu.

8.1.3 Informace o budově

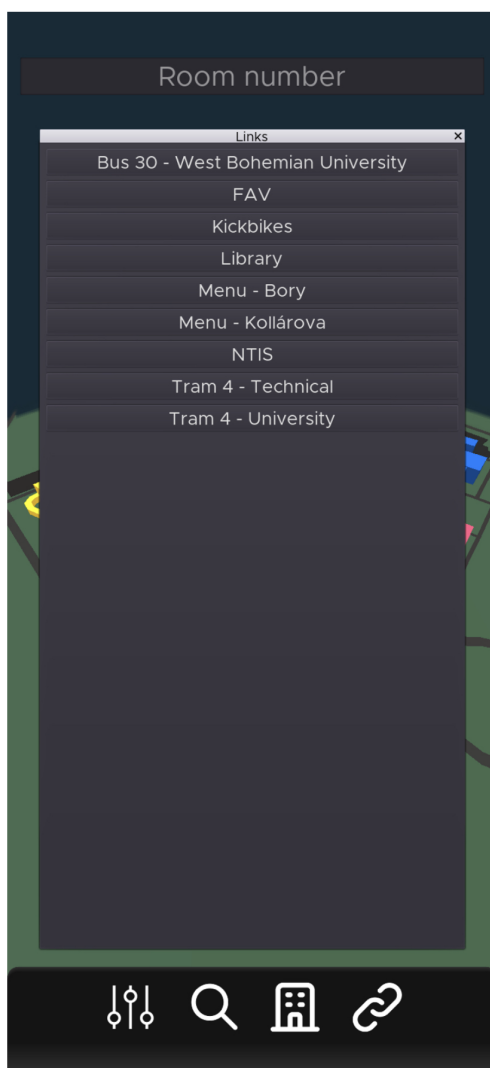
Dále si uživatel může zobrazit informace o budově tak, že klikne na budovu v mapě, nebo jí najde v seznamu budov. Seznam budov se mu otevře po rozkliknutí ikony budovy, dole v navigační liště (obrázek 8.4). V okně *Detail budovy* vidí informace o budově a také fotografie k této budově. Budovu si může na mapě také zobrazit díky tlačítku *Zobrazit budovu*. Budova se následně na mapě rozbliká stejně, jak tomu bylo po stisknutí v okně *Detail místnosti*.



Obrázek 8.4: Okno s detailem budovy. Obsahuje informace o budově, fotografii a linky přiřazené k dané budově.

8.1.4 Odkazy

Další přidaná funkcionalita jsou například odkazy v aplikaci. Uživatel může rozkliknout v navigační liště ikonu odkazu a následně vybrat požadovaný odkaz v seznamu (obrázek 8.5). Odkaz se mu otevře ve výchozím prohlížeči systému iOS. Pokud je odkaz přiřazen k budově, zobrazí se odkazy také v okně *Detail budovy* (obrázek 8.4). Díky tomu, pokud uživatel v mapě klikne například na budovu menzy, tak se mu vedle informací o budově a fotky zobrazí odkazy na jídelníček.



Obrázek 8.5: Okno se všemi odkazy v aplikaci. Po kliknutí na odkaz se otevře okno prohlížeče.

8.2 Knihovny a zdroje

Pro Godot existuje mnoho knihoven (pluginů). Jeden z nich, který jsem využil je například *Godot SQLite*. Jedná se plugin pro podporu SQLite. Ukládání dat do SQLite databáze není jediný způsob ukládání dat v aplikaci. Lze je například ukládat do souboru ve formátu JSON, nebo v jakémkoli vlastním formátu. Já si SQLite databázi zvolil kvůli rychlosti hledání v tabulkách. Repozitář knihovny můžete nalézt na této adrese: <https://github.com/2shady4u/godot-sqlite>

Pro ikony v aplikaci jsem využil volně dostupné ikony *Heroicons*, které jsou dostupné pod licencí MIT. Ke stažení je dostupných přibližně 230 ikon ve dvou různých stylech (obrysy a plné). Exportované ikony jsem ovšem převedl z formátu SVG do formátu PNG, protože se s nimi v Godotu pracuje mnohem lépe. Všechny ikony jsem stahoval z tohoto webu: <https://heroicons.com>

8.3 Animace

Zobrazovat budovu jsem se rozhodl tak, že jí celou "rozblíkám". K tomu jsem využil nativní nástroj v Godotu pro tvorbu animací (Node AnimationPlayer). Po mnoha pokusech jsem se rozhodl, že nejlepší bude animaci spouštět pro samotný model a u něj s časem animace měnit materiál. I když jde pracovat s parametry materiálu napřímo, vyskytovali se tam chyby, které užívání aplikace znepříjemňovaly. Proto jsem rozhodl zduplikoval všechny materiály a u každého změnil barvu na tmavší. Následně se při animaci u objektu měnil celý materiál.

Při kliknutí na jakýkoli jiný objekt, nebo tlačítka v dolním panelu se animace automaticky zastaví. Toho se docílí tak, že se přes signal zavolá funkce `_on_GUI_stop_flashing()` ve skriptu `Map.gd`. Ve funkci se u nodu `FlashPlayer` nastaví čas na 0 a zavolá funkce `stop`. Čas se musí nastavit na 0, jinak by se animace mohla zastavit v době, kdy je u budovy změněn materiál u kterého je změněný shader na tmavší barvu.

8.4 Databáze

Aplikace pro interní data využívá databázi SQLite. V ní se nachází data o budovách a místnostech. Většina dat je uložena již v aplikaci, proto jde aplikaci používat offline okamžitě po nainstalování.

SQLite jsem si vybral pro úložiště kvůli rychlosti a jednoduchosti použití. SQLite je malá knihovna naprogramovaná v jazyce C. Její přednosti jsou rychlost, velikost a to, že dokáže běžet na velké množině zařízení, protože je implementována v běžně používaných operačních systémech [31]. Všechna data databáze jsou uloženy v souboru. S celou databází se dá manipulovat jako se souborem. Toho jsem využil pro zkopírování do uživatelského prostoru (detailněji níže). SQLite je jedním z nejpoužívanějších databázových systémů současnosti. Sice ho nelze použít pro velké aplikace a weby [39], ale kvůli velikosti a možnosti ho spustit na skoro jakémkoli zařízení se využívá v mnoha mobilních aplikacích, IoT, jednoduchých webových aplikacích, nebo počítačových aplikacích. Apple jej často využívá v nativních aplikacích, které se nacházejí v MacOS[35].

Pro správu SQLite databáze jsem využil aplikaci *SQLiteStudio* ve které jsem vytvořil celou databázi, tabulky a vyplnil některá data, která nešla vyplnit přes API. V aplikaci jsem pro práci s SQLite používal speciální plugin pro Godot (více kapitola 8.2). Ukázku funkce s SQL příkazem pro získání odkazů v aplikaci naleznete níže na ukázce kódu 8.3.

Při implementaci jsem narazil na problém platformy iOS, Android a Web. Pokud chci databázi upravovat, tak jí musím nejdříve zkopírovat do *"user://"* (uživatelského) prostoru a až následně jí mohu používat. Pro jiné platformy (Windows, Linux, ...) se toto řešit nemusí a databáze se může upravovat rovnou v *"res://"* [21].

```
1 func get_links():
2     db.open_db()
3
4     db.query("SELECT * FROM " + table_links + " ORDER BY Title_" +
↪ Locale.get_locale() + " ASC;")
5
6     var links = db.query_result
7     db.close_db()
8     return links
```

Kód 8.3: Ukázka funkce s SQL dotazem ve třídě Database.

8.4.1 Použité tabulky

Dále uvedu tabulky SQLite databáze, která se nachází v aplikaci:

1. Buildings - V tabulce se uschovávají data o budovách. Obsahuje jak

název, tak název modelu, tak popisek budovy v češtině a angličtině.

2. Links - Zde se nachází odkazy, které jsou v aplikaci. Tabulka obsahuje jak název v češtině, tak i v angličtině, tak název budovy, ke které patří. To je ovšem nepovinný údaj.
3. Rooms - Tabulka obsahuje všechny stažené místnosti a potřebné informace k nim.
4. Storage - Tabulka je určena pro uschovávání dat aplikace. Aktuálně se využívá pouze pro uložení data poslední aktualizace a lokalizace aplikace, aby se aplikace po restartu spustila ve vybraném jazyku z nastavení.

8.5 Překlady

Kvůli možnému využití aplikace studenty ze zahraničí nebo z výměnného pobytu jsem pro aplikaci vytvořil překlady. Nyní lze využívat aplikaci jak v českém, tak v anglickém jazyce. Po spuštění aplikace se vybere výchozí jazyk telefonu. Pokud je v telefonu nastaven jiný jazyk, jako například italština, tak se aplikace spustí v angličtině. Jazyk se dá případně změnit v *Nastavení* (obrázek 8.6), kde se po kliknutí na požadovaný jazyk změní jazyk celé aplikace.

Překlady jsou uloženy jako CSV soubor v hlavní složce aplikace. Soubor obsahuje tři sloupce. První obsahuje klíč, podle kterého se vybere překlad, další dva obsahují překlad pro vybraný jazyk. Pokud se klíč vyskytne v poli text, tak se automaticky rozpozná a nahradí za překlad podle aktuálně používaného jazyku.

8.6 Aktualizace

Aplikace nabízí možnost aktualizovat místnosti a budovy z API. Pokud uživatel otevře okno s nastavením (obrázek 8.6), může buď změnit jazyk, nebo aktualizovat všechny místnosti a budovy. Pokud by se o to uživatel pokusil bez přístupu k internetu, vyskočí mu malé dialogové okno s informací, že je aktuálně bez internetu.

Během aktualizace se pro místnosti smažou všechny záznamy z databáze a nahradí novými, staženými z API. U budov se aktualizuje poznámka (z API se vrací buď adresa budovy, nebo její název).

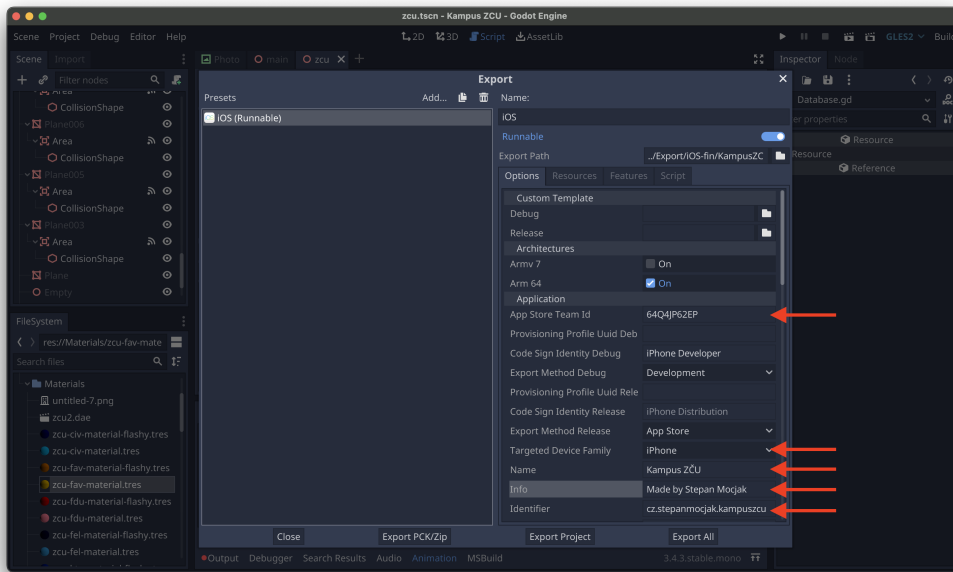


Obrázek 8.6: Okno Nastavení, kde může uživatel změnit jazyk aplikace, nebo aktualizovat data.

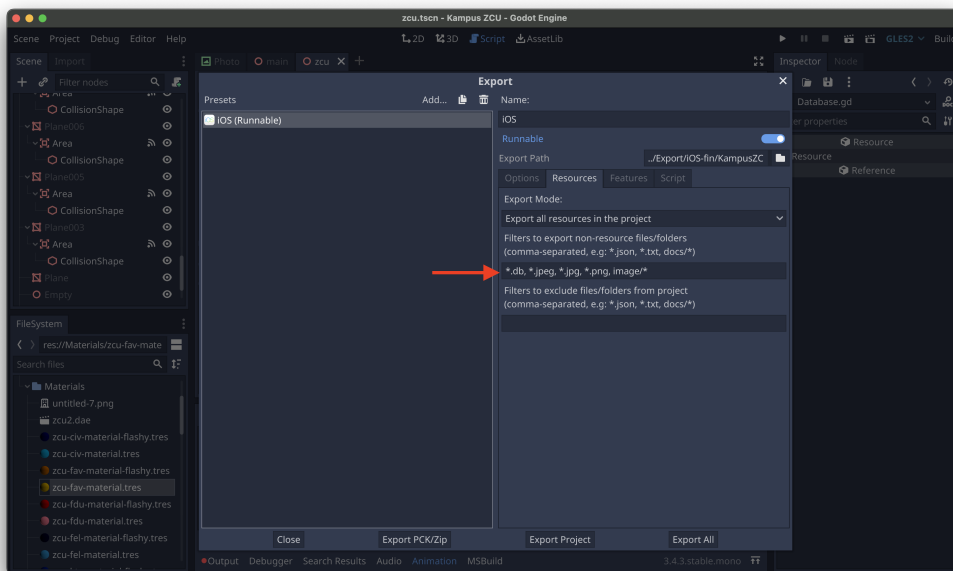
8.7 Export/Instalace

Pro nainstalování aplikaci na telefon je potřeba projekt v Godotu nejprve exportovat pro vybranou platformu (iOS). Aplikaci jsem exportoval standardním způsobem popsáním výše. Pro správnou funkci jsem ale musel změnit několik parametrů. Jeden z potřebných parametrů je například jméno aplikace nebo identifikátor (obrázek 8.7). Godot při exportu ignoruje všechny soubory, které nejsou pro aplikaci důležité. To jsou například zdrojové soubory k ilustracím (*.ai) nebo dokumentace (*.txt, *.docx). Díky tomu se ovšem neimportuje databáze (*.db). Soubory s touto příponou jsem musel

explicitně povolit při exportu (obrázek 8.8).



Obrázek 8.7: Příklad změněných parametrů pro export aplikace z Godotu.



Obrázek 8.8: Explicitní povolení souborů při exportu pro správnou funkčnost aplikace.

Po testování jsem ovšem zjistil, že obdobný problém byl i s obrázky

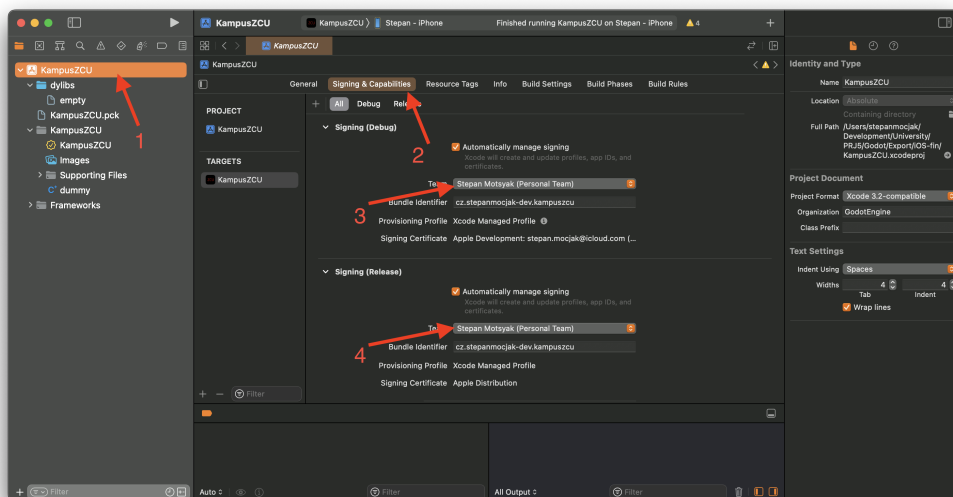
budov. Tam to bohužel nešlo opravit tak snadno. Nakonec jsem musel re-importovat všechny fotografie jako soubory a musel částečně upravit kód, který se staral o zobrazení fotografie.

8.7.1 Xcode

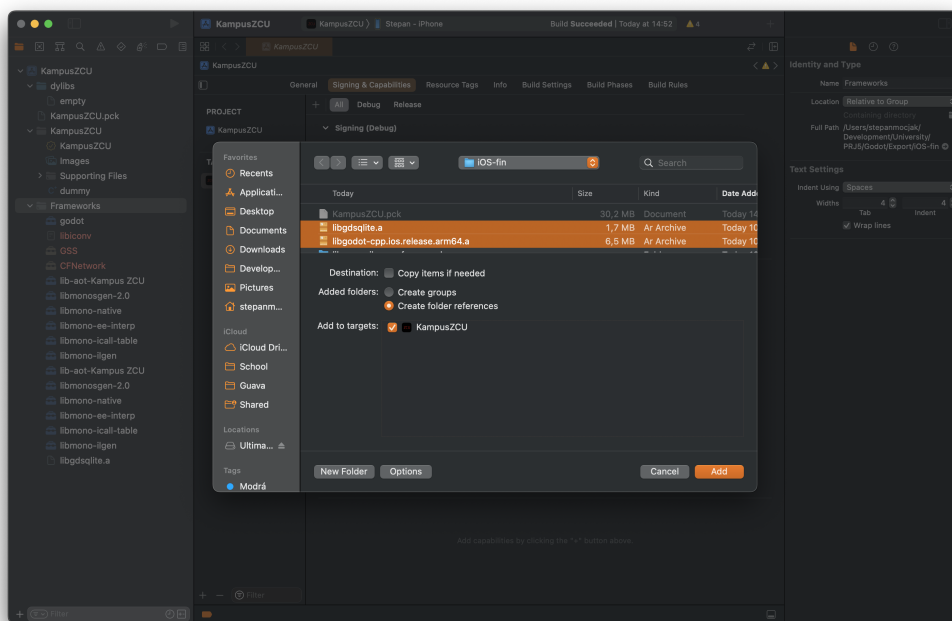
Pro iOS se projekt exportuje jako Xcode projekt. Ten se následně otevře pomocí editoru Xcode. V něm se nejprve musí nastavit správné podpisy (Obrázek 8.9). Pokud se jedná o první export, musí se přidat do exportované složky soubory *libgdsqlite.a* a *libgodot-cpp.ios.release.arm64.a*. Jedná se o knihovny pro plugin *Godot SQLite*, který se využívá pro práci s SQLite databází. Soubory lze v projektu nalézt na této cestě: *KampusZCU-godot/addons/godot-sqlite/bin/ios/armv64*. Následně se musí v Xcode přidat soubory do projektu (obrázek 8.10).

Po předchozích krocích se už jenom musí připojit telefon k počítači se systémem MacOS a v Xcode spustit aplikaci. Pokud se na telefonu nikdy neinstalovala aplikace z Xcode, kterou jste například sami vyvíjeli, musíte povolit váš vývojářský účet (obrázek 8.11).

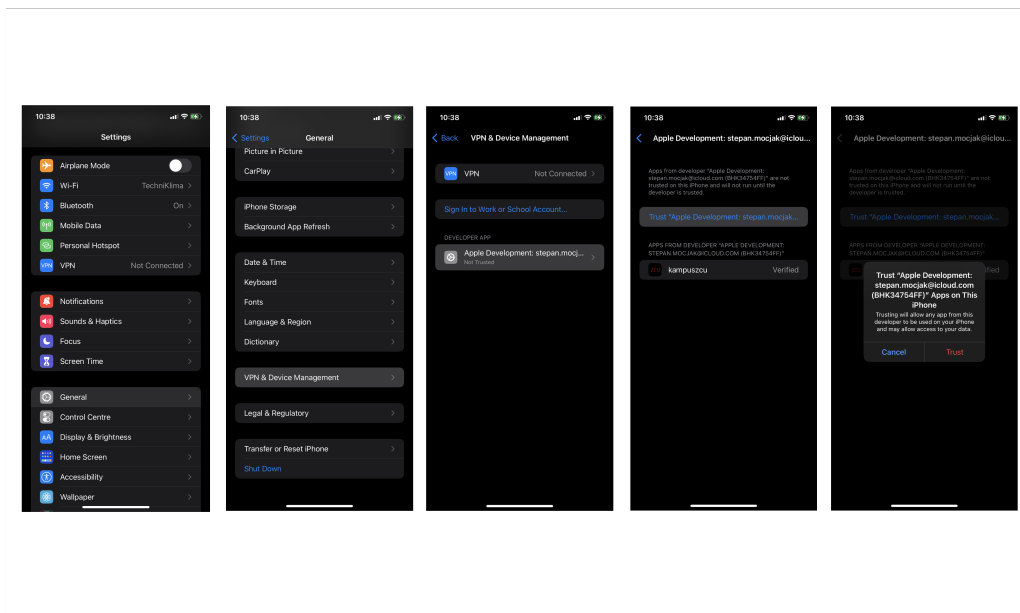
Pro ulehčení instalace na vlastní telefon je celý Xcode projekt již vyexportován a přiložen k projektu. Uživatel tak již nemusí exportovat projekt z Godotu. V exportovaném projektu jsou také již přiloženy knihovny pro práci s SQLite. Uživatel tak bude muset pouze změnit podpisy (obrázek 8.9).



Obrázek 8.9: Pro instalaci z Xcode je nutné změnit certifikáty na vaše vlastní.



Obrázek 8.10: Po exportu z Godotu je nutno přidat knihovny pro práci s SQLite.



Obrázek 8.11: Povolení vašeho vývojářského účtu v systému iOS.

8.8 Nasazení

Aplikaci plánuji publikovat v AppStore. Bohužel pro to musím mít účet vedený pod Apple Developer Programem a to stojí 99 \$ za jeden rok. Proto bych aplikaci prozatím šířil jako open source projekt a každý uživatel, který jí bude chtít používat, si jí nainstaluje přes Xcode. Kroky pro instalaci z vlastního počítače jsou popsány výše.

Pro případné zveřejnění aplikace v AppStore již hledám partnera, který by zveřejnil aplikaci pod svým účtem. Hodlám kontaktovat Centrum informatizace a výpočetní techniky ZČU pro možnou prezentaci a publikaci aplikace přes ně. S kolegou Lukášem Freyem jsme již totiž s CIVem spolupracovali na publikaci webové aplikace *Buddy System*.

8.9 Rozšíření

Jako budoucí rozšíření vidím využití polohových údajů telefonu a GPS. Bohužel prozatím Godot nenabízí žádný takovýto plugin, proto by se musel naprogramovat. Následně by uživatel mohl vidět, kde se přesně nachází.

Společně s polohou vidím možnost navigace v aplikaci jako další krok, kdy uživatel vyhledá učebnu, klikne na navigovat a aplikace mu ukáže nejkratší možnou trasu z bodu, kde se nachází do učebny kterou hledá. Pro tuto funkcionalitu již v Godotu existují *NavigationMesh* pro hledání nejkratší možné trasy.

Další možné rozšíření je propojení aplikace s webovou aplikací *Buddy System*, kterou jsme s kolegou Lukášem Freyem společně vyvinuli pro propojení českých studentů a studentů z programu Erasmus. Aplikace již podporuje angličtinu a je určena pro orientaci v kampusu. Je pravděpodobně, že jí tedy budou zahraniční studenti používat a takto by se jim zjednodušila práce.

Dále plánuji rozšířit aplikaci o více odkazů (obrázek 8.5, které pomůžou studentům při hledání informací v kampusu. Přidání nových odkazů nebude složité. Stačí pouze upravit SQLite databázi.

Plánuji do aplikace také přidat více jazyků. Jedním z nich je například ukrajinština. Jsem rodilý mluvčí a s překlady do ukrajinštiny pro FAV mám již zkušenosti.

Dále bych chtěl aplikaci rozšířit o možnost zobrazit malou mapu (půdorys) podlaží a umístěním místnosti na daném podlaží v detailu místnosti. To by usnadnilo hledání místnosti na patře. Mapa by buď byla v aplikaci uložena jako obrázek, nebo by se muselo upravit API, aby vracelo i půdorys podlaží. V prvním případě by se jednalo o jednoduchou úpravu, protože zobrazování obrázku již funguje u budov. Zvýšila by se tak ovšem paměťová náročnost aplikace, protože by zabírala více místa v telefonu.

9 Testování

9.1 Chování na různých zařízeních

Aplikaci jsem otestoval na mém osobním telefonu - iPhone 11, dvou různých telefonech iPhone SE druhé generace a také rovnou z Godotu na MacBooku Pro 16 2019. iPhone 11 má výřez v obrazovce, kterému se říká notch. Aplikace by se měla korektně zobrazovat na všech telefonech od firmy Apple.

Jelikož se aplikace spouští přes celou obrazovku, tak při vývoji nastal problém u iPhone 11, kdy se horní vyhledávací lišta zobrazovala pod notchem a nešlo tak vidět, co uživatel zadává. Proto jsem lištu přesunul dolů. Bohužel jsem nenašel způsob, jakým detekovat, zda telefon má, nebo nemá notch a kolik místa zabírá. Aplikace se ale nyní korektně zobrazuje jak na iPhone SE bez výřezu v obrazovce, tak na iPhone 11 s výřezem.

9.2 Stabilita aplikace

Aplikace by neměla spadnout po přechodu do jiné aplikace, minimalizaci, nebo v případě, že uživateli někdo při používání zavolá. Aplikace by se měla vrátit do původního stavu.

Aplikaci si zachovává svůj stav, pokud byl její chod přerušen. Například po kliknutí na odkaz se zobrazí prohlížeč. Při vrácení zpět je aplikace ve stejném stavu, jako před kliknutím na odkaz. Stejně se chová při jakémkoli jiném vyrušení (např. telefonním hovoru).

9.3 První spuštění

První interakce uživatele s aplikací je její spuštění. Mohou nastat dva scénáře. Aplikace se spustí s připojením k internetu, nebo bez. Aplikace by se měla v obou případech spustit a neměl by nastat žádný problém.

Testování ukázalo, že rozdíl v chování aplikace nenastává. Obrazovku po spuštění můžete vidět na obrázku 8.2.

9.4 Otočení obrazovky

U aplikace je zakázáno otáčení obrazovky. Při pokusu o otočení by tak nemělo dojít k žádné akci a aplikace by měla zůstat na výšku.

Žádné problémy s otočením jsem při používání nezaznamenal. Dokonce, pokud přejdu z jiné aplikace, která se používá na šířku, tak se aplikace stále zobrazuje na výšku.

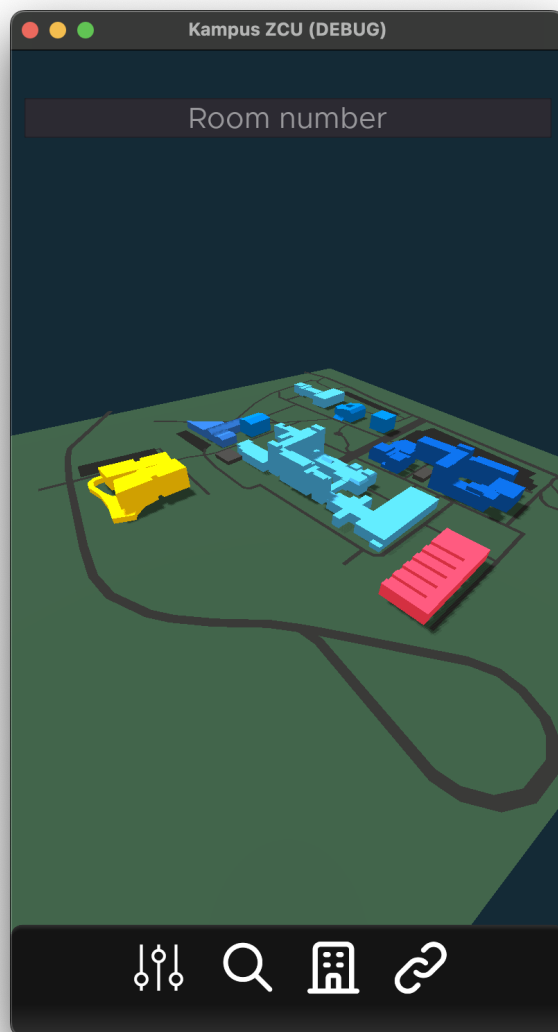
9.5 Stíny v aplikaci

Aplikace měla podporovat stíny. Při vývoji jsem s touto funkcionalitou počítal.

Bohužel při testování na reálném zařízení jsem přišel na chybu, kdy se místo stínů zobrazovali zvláštní artefakty (obrázek 9.1). Pokud jsem aplikaci spouštěl na počítači, žádné potíže se stíny nebyli (obrázek 9.2). Chybu jsem nahlásil na oficiálním Discord serveru Godotu. Podobný problém se vyskytl ještě u jednoho dalšího uživatele. Navrhované možnosti opravy u mě nefungovali a tudíž jsem se po několika dalších pokusech o opravu rozhodl v aplikaci stíny vypnout. Chybu plánuji nahlásit v oficiálním GitHub repozitáři Godotu po tom, co problém podrobněji prozkoumám.



Obrázek 9.1: Artefakty místo stínů. Snímek obrazovky aplikace z iPhone 11.



Obrázek 9.2: Ukázka stínů v aplikaci spuštěné v Godotu na operačním systému MacOS.

9.6 Vyhledání místnosti

Pro vyhledávání se používá horní vyhledávací lišta. Klávesnice by se měla zobrazit buď při kliknutí na lištu, nebo po kliknutí na ikonu lupy na dolní liště. Při zadávání by neměl nastat žádný problém ani pokud nebyla nalezena žádná místnost, ani pokud je veliký počet nalezených místností.

Pro zadání hledané místnosti musí uživatel kliknout buď na horní lištu, nebo na ikonu lupy na dolní liště. Po kliknutí se otevře klávesnice a uživa-

tel může zadávat název hledané místnosti. Při testování jsem nezpozoroval žádné neobvyklé chování. Uživatel může hledanou učebnu (text) jednoduše smazat kliknutím na křížek ve vyhledávací liště. Pro schování klávesnice stačí na klávesnici zmáčknout klávesu *return*.

9.7 Zobrazení místnosti

Po vyhledání místností se uživateli zobrazí list místností, odpovídající hledanému výrazu. Po kliknutí na místnost by se mělo otevřít okno s informacemi o místnosti. Pokud je uživatel připojen k internetu, mělo by se mu zobrazit tlačítko pro otevření rozvrhu jako PDF a pod ním list s rozvrhovými akcemi od současného data dále.

Při testování jsem nezpozoroval žádné neobvyklé chování odlišné od předpokládaného chování. Okno s místnostmi lze zavřít kliknutím mimo okno. Pokud nejsou k místnosti přiřazené některé informace, data se v okně nezobrazí.

9.8 Stažení dat z API

V aplikaci lze stáhnout aktualizovaná data z API. K tomuto účelu je v aplikaci (okno nastavení, obrázek 8.6) tlačítko pro aktualizaci. Po jeho kliknutí by se měli zaktualizovat data místností a budov.

Při prvním testování jsem přišel na chybu. Pokud není dostupný internet, nepovedla se aktualizace a došlo ke smazání všech místností. Chybu jsem opravil a nyní, pokud není dostupný internet a uživatel klikne na tlačítko aktualizovat, zobrazí se mu vyskakovací okno s hláškou o nedostupnosti internetu.

9.9 Zobrazení budov

Informace o budově lze zobrazit buď kliknutím na budovu na mapě, nebo vybráním budovy ze seznamu budov, který se otevře po kliknutí na ikonu budovy v dolní liště. U budovy by se měl zobrazovat název a další informace. Také by se u budov měli zobrazovat fotografie pro názornější reprezentaci a odkazy na zajímavosti spjaté s budovou.

Při testování na jiném telefonu, kdy jsem instaloval aplikaci z exportovaného projektu pro Xcode, jsem si všiml, že u budov se nezobrazují fotografie. To jsem již opravil. Chyba a postup opravy byl popsán v kapitole Export/Instalace.

9.10 Odkazy

V aplikaci se nachází odkazy na různé weby spojené s univerzitou. Jedná se například o jídelníček Menzy. Tlačítka s odkazy by se měli zobrazovat buď v okně detailu budovy, nebo v samostatném okně s odkazy (obrázek 8.5). Po kliknutí na tlačítko by se měl spustit výchozí prohlížeč s otevřeným webem (odkazem).

Po kliknutí na tlačítko s odkazem se otevře okno výchozího prohlížeče s požadovaným webem. Z webu se dá vrátit zpět do aplikace bez potíží. Otestoval jsem prohlížeče Safari, Google Chrome a Brave. Všechny se chovali stejně.

9.11 Změna jazyka

Aplikace nabízí dvě lokalizace, českou a anglickou. Po spuštění by se měl vybrat jazyk telefonu. Uživatel může jazyk změnit v okně *Nastavení* (obrázek 8.6).

Při testování jsem na žádný problém s texty aplikace nenarazil, všechny texty jsou přeloženy do angličtiny a češtiny. Bohužel při změně jazyku a ukončení aplikace se aplikace následně spouštěla vždy v jazyku telefonu. Problém jsem již vyřešil a lokalizace se ukládá.

9.12 Shrnutí testování

Na základě řady testovacích scénářů byla ověřena funkčnost aplikace. Objevené chyby byly opraveny.

10 Závěr

Cílem práce bylo vytvořit aplikaci pro studenty, návštěvy, studenty z programu Erasmus a další, která usnadní orientaci v kampusu Západočeské univerzity. Aplikace vhodně doplňuje rozvrhové aplikace, které již existují a pokrývá skupinu uživatelů s telefony značky Apple s operačním systémem iOS. Aplikace nabízí i některé další funkcionality, které pomohou s hledáním informací. Jedná se například o užitečné odkazy nebo rozvrh místnosti. Při vývoji jsem využil univerzitní zdroje a to jak v podobě API, tak i informací, které jsem dohledal na univerzitním webu, nebo jednotlivých fakultních webech.

Během práce na projektu jsem se seznámil s vývojem aplikací pro iOS. Porovnal jsem různé vývojové nástroje. Vyzkoušel jsem si jak nativní vývoj, tak vývoj aplikací v herním enginu Godot, který jsem si nakonec vybral pro výslednou realizaci. Godot se ukázal použitelný i pro mobilní aplikace. Aplikaci jsem otestoval, detekované chyby opravil a navrhl další možná rozšíření.

Během vývoje jsem zjistil některé chyby, které jsou v Godotu. Ty se hodlám podrobněji prozkoumat a nahlásit v oficiálním repozitáři na GitHub.

Aplikaci plánuji dále rozvíjet a rád bych aplikaci publikoval na App Store. Buď přes účet CIVu, nebo můj osobní.

Slovník

- Android** Mobilní open-source operační systém vyvíjený primárně společností Google.. 12–14, 16–18, 33, 43
- API** Rozhraní pro programování aplikací.. 17, 20, 43, 44, 54, 68
- App Store** Obchod s aplikacemi pro iOS.. 14, 18, 19, 56
- BlackBerry OS** Mobilní operační systém od společnosti BlackBerry.. 12
- DOD** Den otevřených dveří.. 10
- endpoint** URL adresa, kterou obsluhuje server a při volání vrací data.. 20–22
- GitHub** Online uložisko kódu podporující verzování.. 36, 52, 56
- iOS** Mobilní operační systém od společnosti Apple.. 6, 9, 10, 12–18, 22, 28, 32, 33, 37, 41, 43, 45, 47, 56
- iPad** Tablet vyvinutý společností Apple.. 13, 16
- iPhone** Mobilní telefon vyvinutý společností Apple.. 13, 23, 51, 52
- MacOS** Operační systém pro počítače společnosti Apple.. 13, 14, 16, 18, 25, 33, 43, 47, 53
- OS** Operační systém.. 12, 13
- PNG** Rastrový formát pro zobrazování obrázků (fotek).. 42
- SQLite** Malý databázový systém fungující jako malá knihovna napsaná v jazyce C.. 10, 42, 43, 47, 50
- SVG** Vektorový formát pro zobrazování ilustrací (obrázků).. 42
- Swift** Programovací jazyk vyvíjený společností Apple.. 16
- Symbian** Mobilní operační systém od společnosti Nokia.. 12
- UWP** Univerzální platforma Windows.. 33

Windows Mobile Mobilní operační systém vyvíjený Microsoft.. 12

Xcode Vývojové prostředí, primárně určeno na tvorbu aplikací pro zařízení společnosti Apple.. 14, 16, 33, 35, 47, 49, 54, 64

ZIP Souborový formát určený pro kompresi dat.. 34

Literatura

- [1] *Infografika: Jak dlouho Apple podporuje iPhone s aktualizacemi?* [online]. ip-servis.cz, 2020. [cit. 2022/03/13]. Detaily o aktualizacích iOS. Dostupné z: <https://ip-servis.cz/clanky/infografika-jak-dlouho-apple-podporuje-iphone-s-aktualizacemi/>.
- [2] *Apple Developer Enterprise Program* [online]. developer.apple.com. [cit. 2022/01/6]. Enterprise program pro zařízení Apple. Dostupné z: <https://developer.apple.com/programs/enterprise/>.
- [3] *Apple Developer Program Membership Fee Waivers* [online]. developer.apple.com. [cit. 2022/05/16]. členství bez poplatků. Dostupné z: <https://developer.apple.com/support/membership-fee-waiver/>.
- [4] *What is Android?* [online]. Tutorialspoint.com, . [cit. 2022/01/5]. Informace o Androidu. Dostupné z: https://www.tutorialspoint.com/android/android_overview.htm.
- [5] *Android Homepage* [online]. Zebra.com, . [cit. 2022/01/5]. Produkty firmy Zebra Technologies a vysvětlení, proč využívají Android. Dostupné z: <https://www.zebra.com/gb/en/cpn/os-migration/explore-our-solutions/android-home.html>.
- [6] *Blender* [online]. blender.com. [cit. 2022/05/20]. Informace o modelovací aplikaci Blender. Dostupné z: <https://www.blender.org>.
- [7] *C basics* [online]. godotengine.org. [cit. 2022/06/13]. Informace a základy o C v Godotu. Dostupné z: https://docs.godotengine.org/en/stable/tutorials/scripting/c_sharp/c_sharp_basics.html.
- [8] *Choosing a Membership* [online]. developer.apple.com. [cit. 2022/05/16]. Informace o členství ve vývojářském programu. Dostupné z: <https://developer.apple.com/support/compare-memberships/>.
- [9] *Console support in Godot* [online]. godotengine.org. [cit. 2022/06/13]. Exportování projektů na konzole. Dostupné z: <https://docs.godotengine.org/en/stable/tutorials/platform/consoles.html>.
- [10] CONTRIBUTOR, T. *3D model* [online]. techtarget.com. [cit. 2022/05/18]. Co je 3D model. Dostupné z: <https://www.techtarget.com/whatis/definition/3D-model>.

- [11] ETCHEVERRY, I. R. *Introducing C in Godot* [online]. godotengine.org, 2017. [cit. 2022/06/13]. Informace a základy o C v Godotu. Dostupné z: <https://godotengine.org/article/introducing-csharp-godot>.
- [12] *Exporting* [online]. godotengine.org. [cit. 2022/06/13]. Export, podporované platformy a šablony. Dostupné z: https://docs.godotengine.org/en/stable/tutorials/export/exporting_basics.html.
- [13] *Exporting packs, patches, and mods* [online]. godotengine.org. [cit. 2022/06/13]. Exportování rozšíření a popis *.pck souborů. Dostupné z: https://docs.godotengine.org/en/stable/tutorials/export/exporting_pcks.html.
- [14] *Exporting projects* [online]. godotengine.org. [cit. 2022/06/13]. Export celého projektu. Dostupné z: https://docs.godotengine.org/en/stable/tutorials/export/exporting_projects.html.
- [15] *Flutter* [online]. Github.com. [cit. 2022/03/14]. Multiplatformní framework Flutter. Dostupné z: <https://flutter.dev>.
- [16] *Flutter Cube* [online]. Github.com. [cit. 2022/03/14]. Flutter Cube. Dostupné z: https://github.com/zesage/flutter_cube.
- [17] *GDScript basics* [online]. godotengine.org. [cit. 2022/05/15]. Informace a základy GDScriptu. Dostupné z: https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_basics.html.
- [18] *GitHub for Unity editor extension* [online]. Github.com. [cit. 2022/03/15]. Git pugin pro Unity. Dostupné z: <https://unity.github.com>.
- [19] *Godot features* [online]. godotengine.org. [cit. 2022/05/15]. Funkce Godot enginu. Dostupné z: <https://godotengine.org/features>.
- [20] *License* [online]. godotengine.org. [cit. 2022/05/15]. Informace o licenci Godotu. Dostupné z: <https://godotengine.org/license>.
- [21] *godot-sqlite* [online]. Github.com. [cit. 2022/06/17]. Plugin umožňující práci s SQLite v Godotu. Dostupné z: <https://github.com/2shady4u/godot-sqlite>.
- [22] ING. PAVEL POKORNÝ, P. *Blender - naučte se 3D grafiku*. BEN - technická literatura, 2006. ISBN 80-7300-203-5.
- [23] INVERITA. *Flutter vs React Native vs Native: Deep Performance Comparison* [online]. Medium.com, 2020. [cit. 2022/03/14]. Porovnání rychlostí Flutter, React Native a Nativní aplikace. Dostupné z: <https://medium.com/swlh/flutter-vs-react-native-vs-native-deep-performance-comparison-990b90c11433>.

- [24] INVERITA. *Unity vs. Unreal – Choosing a Game Engine* [online]. gamedevacademy.org, 2021. [cit. 2022/03/15]. Porovnání Unreal Engine a Unity. Dostupné z: <https://gamedevacademy.org/unity-vs-unreal/>.
- [25] *Ionic* [online]. <https://ionicframework.com>. [cit. 2022/03/14]. Ionic Framework. Dostupné z: <https://ionicframework.com>.
- [26] *Develop Apps for iOS* [online]. developer.apple.com. [cit. 2022/03/14]. Apple Swift Playground. Dostupné z: <https://developer.apple.com/swift-playgrounds/>.
- [27] JERGA, F. *What Is The Unity Game Engine- All You Need To Know* [online]. Medium.com. [cit. 2022/03/15]. Popis Herního enginu Unity. Dostupné z: <https://medium.com/eincode/what-is-the-unity-game-engine-all-you-need-to-know-d4ce77a1b7d2>.
- [28] KENTON, W. *What Is Apple iOS?* [online]. investopedia.com, 2021. [cit. 2022/01/6]. Co je Apple iOS. Dostupné z: <https://www.investopedia.com/terms/a/apple-ios.asp>.
- [29] LACKO, L. *Vývoj aplikací pro iOS*. Computer Press, 2018. ISBN 978-80-251-4942-3.
- [30] LANKINEN. *Xcode (SceneKit) vs. Unity* [online]. medium.com, 2020. [cit. 2022/03/14]. Porovnání Xcode a Unity pro vývoj aplikací. Dostupné z: <https://lankinen.medium.com/xcode-scenekit-vs-unity-c613beb07f76>.
- [31] *Most Widely Deployed and Used Database Engine* [online]. sqlite.org. [cit. 2022/06/16]. Ukázka platform na kterých běží SQLite. Dostupné z: <https://www.sqlite.org/mostdeployed.html>.
- [32] NAT – FRIENDS. *Google Earth’s Incredible 3D Imagery, Explained* [online]. youtube.com, 2017. [cit. 2022/05/18]. Jak Google Earth generuje 3D modely budov. Dostupné z: https://youtu.be/suo_aUTUpps.
- [33] *Nodes and Scenes* [online]. godotengine.org. [cit. 2022/06/13]. Vysvětlení fungování Nodů a Scén v enginu Godot. Dostupné z: https://docs.godotengine.org/en/stable/getting_started/step_by_step/nodes_and_scenes.html?highlight=nodes.
- [34] *What is Android* [online]. Android.com. [cit. 2022/01/5]. Oficiální představení Androidu. Dostupné z: <https://www.android.com/what-is-android/>.

- [35] PANCHAL, K. *Everything you need to know about SQLite Mobile database* [online]. ourcodeworld.com, 2019. [cit. 2022/06/16]. Popis kde využívat SQLite a kde se již využívá. Dostupné z: <https://ourcodeworld.com/articles/read/737/everything-you-need-to-know-about-sqlite-mobile-database>.
- [36] *Pixelorama* [online]. Github.com. [cit. 2022/06/6]. Aplikace vytvořená v enginu Godot. Dostupné z: <https://github.com/Orama-Interactive/Pixelorama>.
- [37] *Godot Python, because you want Python on Godot !* [online]. Github.com. [cit. 2022/05/15]. Python pro Godot engine. Dostupné z: <https://github.com/touilleMan/godot-python>.
- [38] RICHTER, T. *Aktualizace Androidu? Někdy až výsměch uživatelům* [online]. investopedia.com, 2020. [cit. 2022/03/12]. Popis aktualizací systémů Android. Dostupné z: <https://mobilizujeme.cz/clanky/aktualizace-androidu-nekdy-az-vysmech-uzivatelum>.
- [39] S., E. *SQLite vs MySQL – What’s the Difference* [online]. hostinger.com, 2022. [cit. 2022/06/16]. Porovnání SQLite s jiným databázovým systémem (MySQL). Dostupné z: https://www.hostinger.com/tutorials/sqlite-vs-mysql-whats-the-difference/#Pros_and_Cons_-_SQLite_vs_MySQL.
- [40] *CS193p - Developing Apps for iOS* [online]. stanford.edu, 2021. [cit. 2022/03/14]. Lekce Standfordské univerzity ohledně vývoje pro iOS. Dostupné z: <https://cs193p.sites.stanford.edu>.
- [41] *Swift* [online]. developer.apple.com. [cit. 2022/03/14]. Popis programovacího jazyka Swift. Dostupné z: <https://developer.apple.com/swift/>.
- [42] *Learn to code with Swift Playgrounds* [online]. developer.apple.com, 2020. [cit. 2022/03/13]. Apple Swift Playground. Dostupné z: <https://developer.apple.com/swift-playgrounds/>.
- [43] *Unity* [online]. unity.com. [cit. 2022/03/15]. Cenové plány Unity. Dostupné z: <https://store.unity.com/compare-plans>.
- [44] WALFORD, A. *What is Photogrammetry?* [online]. photogrammetry.com, 2017. [cit. 2022/05/18]. Vysvětlení fotogrametrie. Dostupné z: <https://www.photogrammetry.com>.
- [45] *WebKit* [online]. developer.apple.com. [cit. 2022/03/14]. Popis Webového jádra WebKit. Dostupné z: <https://developer.apple.com/documentation/webkit>.

- [46] *Xcode 13* [online]. developer.apple.com. [cit. 2022/03/14]. Popis vývojového prostředí Xcode. Dostupné z: <https://developer.apple.com/xcode/>.

Přílohy

- Instalace
- Uživatelská příručka
- Struktura přiloženého ZIP souboru

Instalace

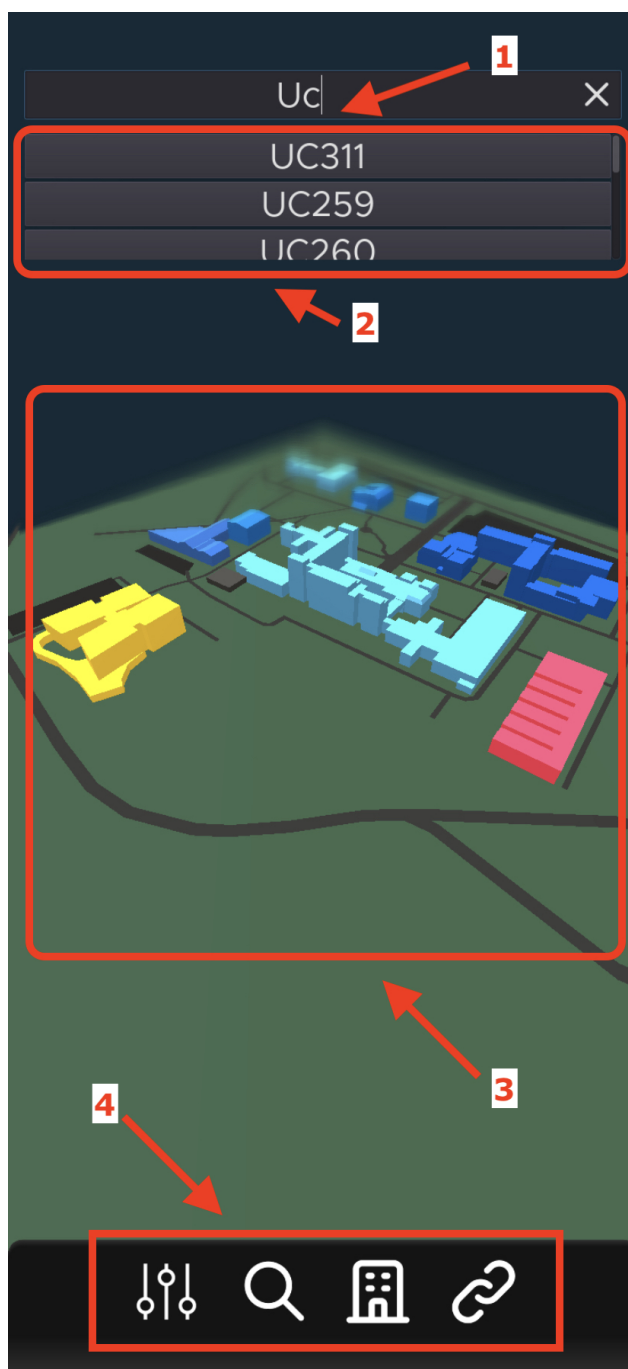
Pro instalaci se uživateli nabízí dvě možnosti. Může aplikaci nejdříve vyexportovat z projektu v Godotu (jedná se o složitější proces) a následně nainstalovat přes Xcode. Nebo, může použít přibalený vyexportovaný projekt pro Xcode a aplikaci nainstalovat rovnou do zařízení. Jelikož je druhá možnost pro koncového uživatele o mnohem jednodušší proces, doporučuji instalaci na koncový telefon tímto způsobem. Podrobnější popis instalace se dočtete v kapitole 8.7.

Uživatelská příručka

Po nainstalování aplikace se na ploše uživatele objeví nová ikona (obrázek 10.1) s názvem *Kampus ZČU*. Po stisknutí se spustí aplikace a uvidí hlavní obrazovku aplikace (obrázek 10.2). Zde vidí mapu kampusu reprezentovanou 3D modelem. S mapou může uživatel rotovat.



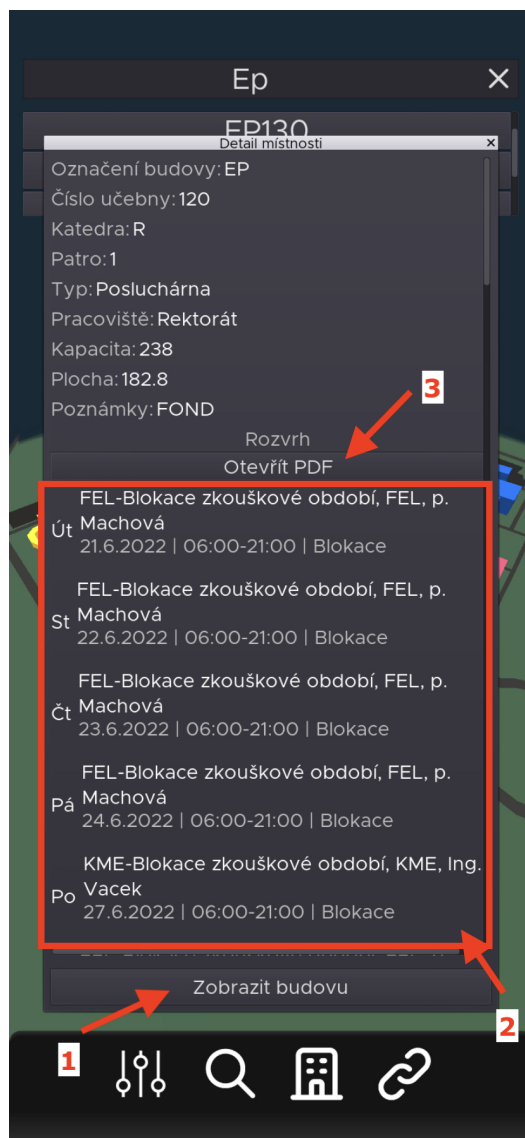
Obrázek 10.1: Ikona aplikace na ploše telefonu.



Obrázek 10.2: Hlavní obrazovka aplikace. 1 - Horní lišta s polem pro hledání místností. 2 - Seznam s vyhledanými místnostmi. 3 - 3D model s mapou kampusu. 4. Dolní lišta s tlačítky: Nastavení, Hledat, Budovy, Odkazy

Pro hledání místností může uživatel buď kliknout na ikonu lupy v dolní liště, nebo na horní lištu, kam zadá název hledané místnosti. Při zadávání funguje našeptávač, uživatel tak nemusí zadat celý název, ale pouze ho vy-

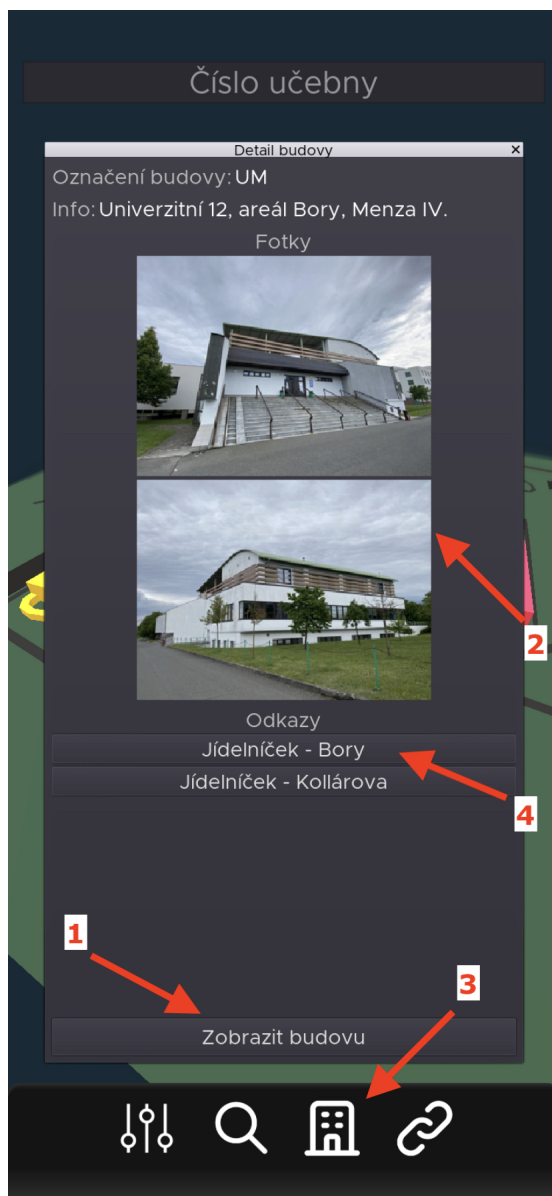
brat ze seznamu pod horní lištou. Po kliknutí na místnost ze seznamu se uživateli zobrazí detail místnosti. Zde uvidí informace o místnosti, v jaké budově se nachází a na jakém patře. Budovu si odsud jednoduše zobrazí stisknutím tlačítka *Zobrazit budovu*, které se nachází dole v dialogu. Zde se také může podívat na akce, které se v místnosti budou konat, nebo si zobrazit kompletní rozvrh místnosti v PDF. Ukázka dialogu na obrázku 10.3. Pro skrytí okna s detailem místnosti stačí uživateli kliknout mimo okno.



Obrázek 10.3: Popis okna s informacemi o místnosti. 1 - Tlačítko pro zobrazení budovy na mapě. 2 - Rozvrhové akce, které se budou v místnosti konat. 3 - Všechny rozvrhové akce ve formátu PDF.

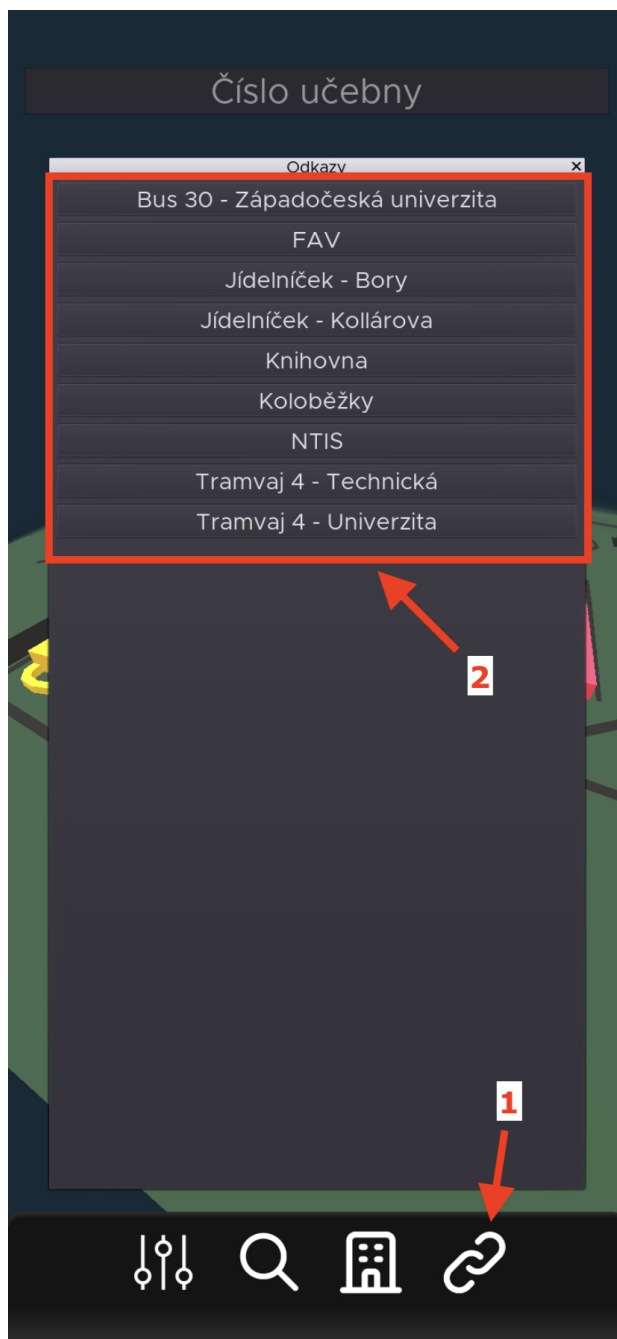
Zobrazit detail budovy může uživatel dvěma způsoby. Buď klikne na bu-

dovu na mapě, nebo jí najde v seznamu budov, který se otevře po kliknutí na ikonu budovy v dolní liště. V okně uvidí název budovy a informaci staženou z API. Také zde uvidí fotografii k budově. Ta se zobrazí, pouze pokud existuje. Uživatel má, obdobně jako v případě místnosti, možnost zobrazit budovu na mapě. A také, pokud jsou k budově přiřazené webové odkazy, tak se mu v detailu zobrazí. Okno *Detail budovy* můžete vidět na obrázku 10.4.



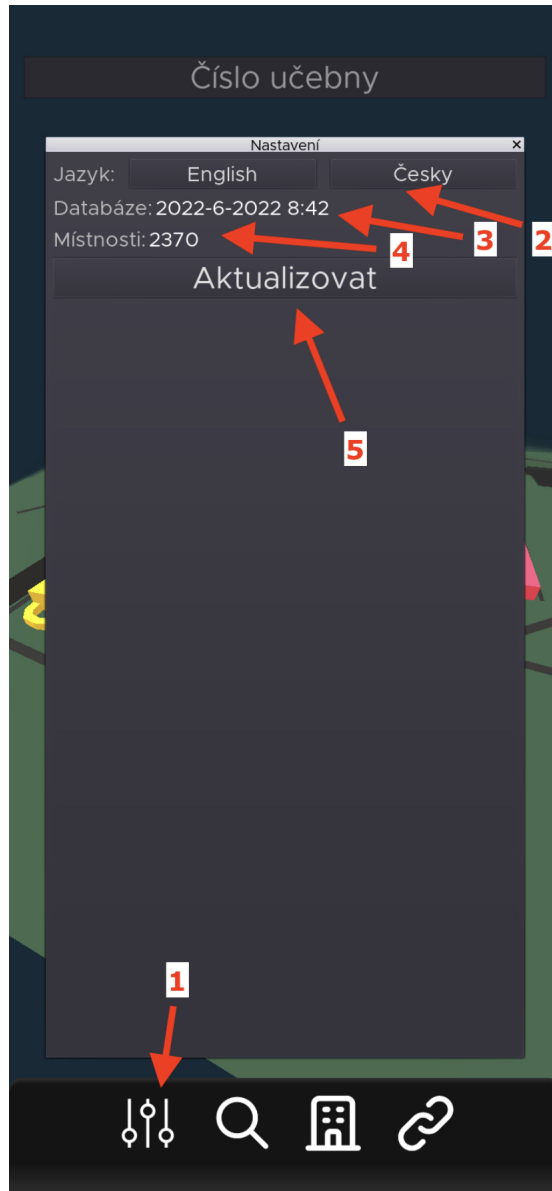
Obrázek 10.4: Okna s informacemi o budově. 1 - Tlačítko pro zobrazení budovy na mapě. 2 - Fotografie budovy. 3 - Tlačítko pro zobrazení seznamu všech budov. 4 - Odkazy přiřazené k budově

Uživatel může zobrazit okno se všemi užitečnými odkazy v aplikaci stisknutím tlačítka s ikonou odkazu (řetězu) na dolní liště (obrázek 10.5). Stejně tak, pokud je odkaz přiřazen k budově, můžete ho vidět i v detailu budovy (obrázek 10.4).



Obrázek 10.5: Okna s odkazy v aplikaci. 1 - Tlačítko pro zobrazení okna se všemi linky 2 - Všechny odkazy v aplikaci.

Pro správu aplikace existuje v aplikaci nastavení. To se otevře kliknutím na ikonu nastavení v dolní liště. Zde si uživatel může změnit jazyk telefonu a případně aktualizovat místnosti. Uvidí zde například čas poslední aktualizace a počet místností aktuálně uložených v databázi. Pokud se u databáze zobrazuje místo data nápis *Offline*, tak aplikace používá výchozí databázi, která je přidaná do aplikace a ještě nebyla aktualizována. Popsaný snímek obrazovky můžete vidět na obrázku 10.6.



Obrázek 10.6: Okno nastavení. 1 - Tlačítko pro zobrazení okna s nastavením. 2 - Tlačítka pro přepnutí jazyka telefonu. 3 - Datum poslední aktualizace. 4 - Počet místností v databázi. 5 - Tlačítko pro aktualizaci databáze.

Struktura přiloženého ZIP souboru

1. Text_prace - Složka bakalářská práce ve formátu LaTeX. Zde se nachází i BP.pdf.
2. Aplikace_a_knihovny - Zdrojové kódy a aplikace.
 - (a) KampusZCU-godot - Zdrojové kódy. Projekt v Godotu.
 - (b) iOS-fin.zip - Exportovaný projekt pro Xcode.
3. Readme.txt - Popis adresáře ZIP.