



Technická dokumentace k softwaru projektu č. VI20172020068

Nástroje a metody zpracování videa a obrazu pro zvýšení efektivity operací bezpečnostních a záchranných složek (VRASSEO)

DroCo – Řídící a Vizualizační Nástroj pro Více-dronové Mise

Ing. Daniel Bambušek; Bc. Róbert Hubinák; Ing. Vítězslav Beran, Ph.D.

Fakulta informačních technologií

Vysoké učení technické v Brně

Božetěchova 1/2

612 66 Brno, Česko

prosinec, 2020

Obsah

| | |
|---------------------------------|----|
| Technická dokumentace | 3 |
| Motivace, principy a cíle | 3 |
| Architektura systému | 4 |
| Podporované drony | 5 |
| Programátorský manuál | 7 |
| Architektura aplikace | 7 |
| Definice misí | 8 |
| Server pro sdílení letových dat | 9 |
| Uživatelský manuál | 11 |
| Základní nastavení | 11 |
| Rozhraní a uživatelské prvky | 12 |
| Instalační manuál | 14 |
| Reference | 15 |

Technická dokumentace

Dokumentace popisuje vzniklou SW aplikaci s GUI (DroCo – *Multi-Drone Control Visualization Tool*) cílenou na vizualizaci a vzdálené efektivní řízení dronu při více-dronových misích a kooperacích. Aplikace využívá princip rozšířené virtuality, kdy do offline dat (mapy, 3D modely budov) vkládá online data ze senzorů dronu za účelem snížení mentální zátěže pilota. Pilot není limitován obrazem z kamery, může se svobodně rozhlédnout po okolí dronu a získat tak alespoň hrubou představu o prostředí, překážkách atp. (Sedlmajer, 2019, Vienna). V následujícím textu jsou vytyčeny cíle a motivace vzniku této aplikace, její architektury a podporovaných dronů.

Motivace, principy a cíle

Složitější situace, kdy je např. potřeba vzdálené operace s horší viditelností nebo provádění úkonů ve složitém prostředí, mnohdy vyžadují zkušené piloty. V některých situacích vyžadují mise i kooperaci více dronů. V takových případech roste jak mentální zátěž pilotů, tak míra rizika ztráty dronu nebo způsobení škody na okolním prostředí. Pilot je často nucen řídit drona s omezeným vizuálním kontaktem a je odkázán na video přenos z kamery dronu (FPV – First Person View). Nárůst rizika a mentální zátěže pilota je mimo jiné způsobena zhoršenou orientací v prostředí.

Tato práce cílí na řešení problému absence znalosti blízkého okolí dronu mimo přímý záběr kamery. K tomuto je navrženo využít dostupné 3D modely prostředí v místě operace s dronem. Zdroje takových 3D modelů mohou být dva: existující off-line 3D model prostředí z dostupných služeb (např. Google maps, Seznam mapy, nebo Open-street maps) a v reálném čase rekonstruovaný hrubý online 3D model okolního prostředí. Pilot tak získá možnost se „rozhlédnout“ po okolí dronu, aniž by musel měnit jeho polohu nebo polohu druhé pomocné kamery, a získat tak hrubé povědomí o objektech kolem dronu.

Toto řešení by ale nemělo znemožnit vidět přímo živá obrazová data z kamery na dronu. Toto je navrženo řešit pomocí rozšířené virtuality, kde se do takto vytvořené 3D scény zobrazí uživateli přímo i živý obraz z kamery na dronu (FPV). Tento zdroj informací je totiž v dané situaci nejpřesnější, informačně zcela zásadní a funguje i v případě, kdy off-line 3D model neobsahuje aktuální modely nebo kdy rekonstruovaný on-line 3D model selže.

Hlavním cílem je snížit mentální zátěž pilota při vzdálených manévrech, kdy musí neustále sledovat své okolí. Navržený princip výše sice nabízí hrubou znalost tohoto okolí i bez nutnosti manévrování s dronem, ale stále ho nutí se „rozhlížet“. Tento problém je řešen s využitím nových GUI prvků. Tyto prvky by měly srozumitelně informovat pilota o okolí dronu, aniž by se musel neustále rozhlížet, nebo si pamatovat, co a jak daleko je za dronem, vedle dronu nebo nad a pod dronem. Příkladem je nově vyvinutý detektor vzdálenosti k nejbližším objektům, který informuje pilota o vzdálenosti k okolí a intuitivně vizualizuje případné nebezpečné přiblížení k blízkým objektům (viz obrázek 1.).

Kromě řady běžných GUI prvků, jako jsou nebezpečné nebo povolené zóny letu nebo řídicí body v misi, které je potřeba vhodně vizualizovat jak v 3D scéně, tak např. v head-up 2D mapě, je potřeba uživateli intuitivně a efektivně zprostředkovat i informace o dalších klíčových objektech celé mise. Těmito objekty jsou především další drony kooperující při řešení mise.



Obr. 1: Ukázka z testovacího letu experimentální mobilní platformou¹. Pilotu je přenášěn živý obraz z kamery dronu, informace o výšce (nadmořská, nad zemí) a rovněž mu jsou zobrazovány vizuální prvky důležité pro splnění mise (označení cíle, vzdálenosti k cíli, ostatní drony podílející se na misi, vzdálenosti k překážkám atd.)

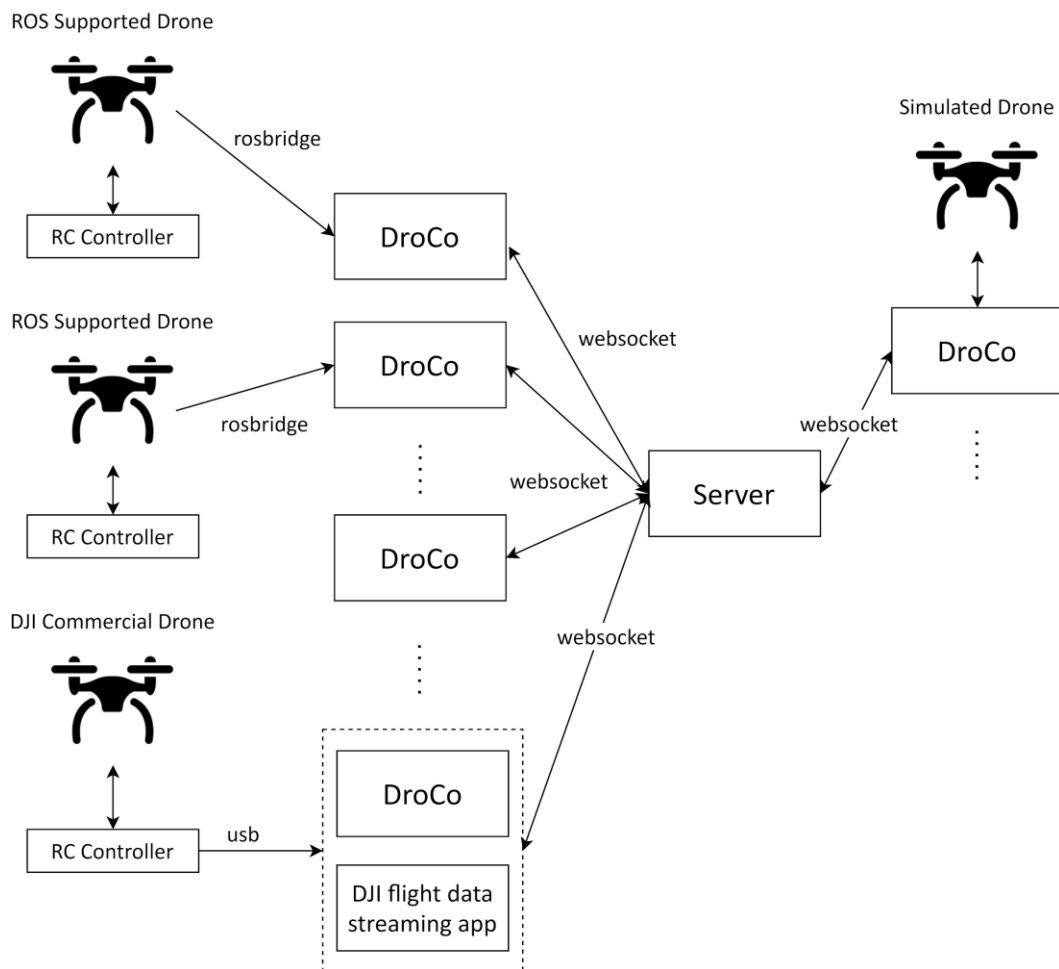
Architektura systému

Stavebním kamenem systému je virtuální model mapy prostředí, ve kterém se pilot nachází. Mapa tvoří scénu a slouží jako metrika, na základě které se přepočítávají reálná GPS data z dronu na polohu modelu dronu ve virtuální scéně. Model mapy je tvořený základním terénem oblasti, doplněným o jednoduché modely budov. Uživatel si může zvolit typ terénu, kde má na výběr terén tvořený satelitními snímky nebo klasickou mapu. Modely budov jsou poměrně nepřesné, kvalitnější textury budov ve většině případů nejsou dostupné, k základní orientaci jsou však postačující. Do scény je při spuštění aplikace vložený model představující ovládaný dron. Aby bylo možné zobrazit model dronu co nejpřesněji, aplikace musí mít v každém momentě k dispozici informaci o poloze a rotaci dronu. Pro zobrazení i FPV (first person video) z dronu do 3D scény je samozřejmě nutné mít k dispozici video-stream z kamery dronu. GPS data jsou po získání ze senzorů dronu přepočítané na souřadnice virtuální scény a na základě této polohy se virtuální dron po scéně pohybuje. Zároveň je video z kamery dronu zobrazované na plátně před modelem dronu (FPV obraz). Kromě polohových dat a dat z kamery aplikace dokáže pracovat mapou obsazenosti, což je 3D pravidelná struktura nesoucí informaci o výskytu překážek v reálné scéně. Mapa obsazenosti se počítá přímo na dronu z mračna bodů získaného stereo-kamerou. Mapu obsazenosti aplikace zobrazuje přímo do virtuální scény a díky těmto informacím může aplikace upozorňovat pilota na blížící se překážky.

Aplikace rovněž obsahuje systém misí. Ten umožňuje pilotovi umístit do scény navigační body, geozóny, které mohou označovat oblasti, kterým je třeba se vyhnout (např. letiště), nebo naopak oblast, ve které se má dron pohybovat (např. průzkum oblasti). Zároveň tento systém umožňuje do scény přidat další drony, které mohou být buď simulované, nebo řízené online reálnými daty z jiných dronů. Ovládat je však možné pouze primární dron. Pro připojení a zobrazení dronů řízených online daty z jiných reálných dronů je potřeba serveru. Každý takový dron potřebuje svoji vlastní instanci aplikace DroCo, která při úspěšném připojení k serveru konstantně posílá letová data na server a ten je následně rozesílá

¹ Testovacím letem se většinou myslí "ručně" nesené zařízení tak, aby bylo možné pořizovat co nejrelevantnější data bez nutnosti reálného letu. Pokud byl realizován reálný testovací let, vždy byl let proveden "na lanku".

všem připojeným klientům. Schéma architektury systému DroCo pro více-dronovou interakci je zobrazeno na obrázku 2.



Obr. 2: Architektura systému DroCo pro více-uživatelskou interakci a kooperaci při plnění misí. Jednotliví klienti (plně simulované, reálné s využitím ROS, DJI Drony apod.) sdílejí spolu navzájem data s využitím serveru.

Uživatelské rozhraní aplikace je navrženo tak, aby vyžadovalo co nejmenší vstup od uživatele. Pomocí grafického rozhraní jsou uživatelům zobrazovány prvky jako navigační body, další drony v misi, směr a vzdálenost nejbližších překážek v okolí dronu apod. Zároveň je rozhraní doplněno o minimapu spolu s navigačním panelem, který zobrazuje důležitá data o misi (zadání aktuální úlohy, vzdálenost a úloha ostatních dronů).

Podporované drony

Aplikace DroCo byla primárně vyvíjena a testována na experimentální mobilní platformě (Plascencia, 2020, Brno) (obr. 3). V aplikaci bylo využito dostupných online dat z dronu (GPS poloha, nadmořská výška, data z gyroskopu, obraz z kamery a rovněž i zrekonstruovaný online model prostředí pomocí stereokamery dronu). Testový let s tímto dronem lze vidět na obrázku 1. Jelikož dron běží na robotickém operačním systému (ROS) a komunikuje s aplikací pomocí *rosbridge* rozhraní, lze pro spojení s aplikací DroCo využít jakýkoliv dron, který využívá a dokáže komunikovat přes ROS (např. DJI Matrice 200, 300 a 600).

Mimo drony využívající komunikaci prostřednictvím ROSu byla prozkoumána možnost integrace dronů, které toto komunikační rozhraní nemají. Konkrétně komerční drony společnosti DJI – Mavic a

Spark. Přímá integrace těchto dronů není zdaleka triviální, neboť neexistují potřebné nástroje (SDK) pro získání online dat z dronů přímo do prostředí Unity, ve kterém byla aplikace vyvíjena. Problém chybějících vývojářských nástrojů je řešen vytvořením pomocné aplikace v Android Studiu (pro které už DJI oficiální knihovny a podporu má), která získává online data z dronu (GPS poloha, výška, gyroskop) a přeposílá je na pomocný server, který je využit pro sdílení letových dat dronů při více-dronových misích.



Obr. 3: Experimentální mobilní platforma využívaná pro vývoj a testování nástroje DroCo.

Programátorský manuál

Výsledkem je aplikace cílená na platformy Android (tablet/mobil, který využívá pilot), Windows a Linux (base station, vhodná pro velitele mise). Aplikace byla vyvinuta v nástroji Unity 3D a její zdrojové soubory, včetně zkompileovaných a spustitelných binárních souborů, jsou zveřejněny na githubu výzkumné skupiny Robo@FIT². Následuje popis jednotlivých prvků aplikace, postup definice misí a popis jednoduchého serveru pro sdílení letových dat, pro případy kooperace více dronů.

Architektura aplikace

Aplikace je rozdělena do několika základních stavebních kamenů (třídy a objekty scény), kde každý zajišťuje danou funkcionalitu (viz obr. 4).

- Objekt **Map**

Vytváří mapu prostředí. Obsahuje třídu *AbstractMap*, která implementuje metody pro přepočítání GPS souřadnic dronu na polohu ve 3D scéně. Třída *MapController* obstarává nastavení vstupních dat pro správné vygenerování mapy. Objekt *Map* spolu s třídou *AbstractMap* jsou součástí použitého pluginu *MapBox*³ pro zobrazování offline mapových dat.

- Objekt **DroneObject**

Představuje model dronu. Jeho třída *DroneController* obstarává zpracování GPS souřadnic a změnu polohy modelu dronu v reálném čase. Tato data, spolu s ostatními zpracovanými daty, jsou získána pomocí rozhraní *rosbridge* a knihovny *RosSharp*⁴. Přesnější popis architektury ROS a propojení s aplikací je popsáno v diplomové práci *Uživatelské rozhraní pro řízení dronu s využitím rozšířené virtuality* (Sedlmajer, 2019, FIT VUT). Zároveň implementuje funkcionalitu pro simulované ovládání dronu. Třída *RayCastHandler* implementuje detekci blízkého se nebezpečí pomocí vrhání paprsků (*raycasting*) směrem od modelu dronu do osmi směrů od něj. Na základě kolizí těchto paprsků s objekty 3D scény jsou postupně aktivované a deaktivované navigační prvky zobrazující nebezpečí v okolí dronu. Jako dceřiné objekty obsahuje sadu hlavních kamer scény a virtuální plátno s obrazem kamery dronu. Tyto obsahují podpůrné třídy pro rotaci kamer a nastavení velikosti plátna.

- Objekt **MainCanvas**

Reprezentuje uživatelské rozhraní aplikace. Jeho dceřiné objekty jsou konkrétně prvky uživatelského rozhraní. O funkcionalitu každého z těchto prvků se stará třída *GuiController*.

- Objekt **OccupancyMapHandler**

Třída *PointCloudSubscriber* načítá mračno bodů (*PointCloud*), ze kterých následně vytváří mapu obsazenosti (3D model typu *mesh*), který je zobrazený do scény. Získání těchto dat probíhá rovněž přes komunikační rozhraní *rosbridge*.

- Objekt **MissionHandler**

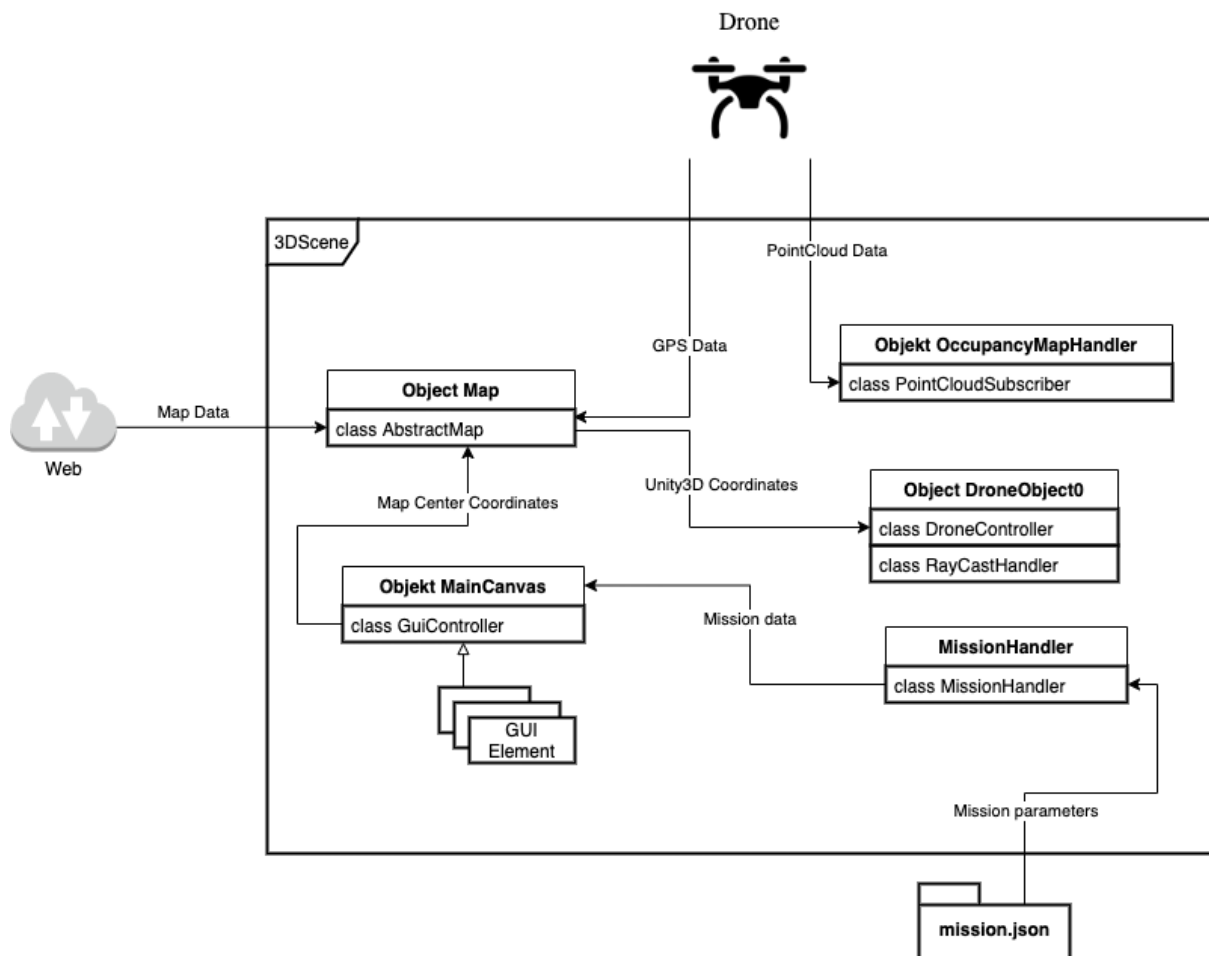
Objekt zpracovává parametry zadané mise a vykresluje jednotlivé prvky do scény. Po vytvoření všech objektů řídí celou misi a pohyb simulovaných dronů. Zároveň řídí

² https://github.com/robofit/drone_vstool

³ <https://www.mapbox.com/unity>

⁴ <https://github.com/siemens/ros-sharp>

polohu HUD elementů označujících jednotlivé prvky mise. Misi je možné definovat v souboru *DroCo_Data/StreamingAssets/mission.json* prostřednictvím JSON formátu.



Obr. 4: Schéma architektury aplikace DroCo.

Definice misí

Mise má dvě základní pole parametrů, pole **checkpoints**, které reprezentuje waypointy a zóny v misi a pole **drones**, které reprezentuje drony podílející se na misi. Jednotlivé checkpoints jsou definované pěti parametry:

- **name** – Jednoznačný identifikátor checkpointu.
- **type** – Určuje typ checkpointu. Podporované jsou 3 základní typy:
 - *regular* – Jednoduchý waypoint, který je považovaný za splněný po tom, co se k němu dostanou všechny přidělené drony.
 - *confirm* – Waypoint s úlohou, kterou musí pilot ručně potvrdit. Je splněný po tom, co se k němu dostanou všechny přidělené drony a pilot potvrdí splnění úlohy.
 - *zone* – Geozóna, reprezentuje zónu, do které se mají drony dostat nebo které se mají vyhnout.
- **points** – Označuje pole bodů reprezentovaných zeměpisnou šířkou (*latitude*), délkou (*longitude*) a výškou nad zemí (*height*), které určují polohu checkpointů na mapě. Pro typy *regular* a *confirm* se jako poloha waypointu uvažuje první bod pole, pro typ *zone* jsou pak body interpretovány jako hrany zóny.
- **drones** – Pole identifikátorů dronů, kterým byl přidělený daný checkpoint. Pokud je pro typ *zone* toto pole prázdné, zóna je považována za zónu, které se mají drony vyhnout.

- **description** – Reprezentuje popis úlohy pro typ *confirm*. Pro ostatní typy je tento parametr volitelný.

Jednotlivé drony podílející se na misi jsou reprezentované čtveřicí **name**, **latitude**, **longitude**, **url**, kde *name* je jednoznačný identifikátor dronu a dvojice *latitude*, *longitude* jsou souřadnice bodu, kde se bude dron nacházet při startu mise. Parametr *url* je nepovinný a umožňuje zadat adresu serveru, kde lze přes API komunikovat aktuální polohu dronu. Pro podrobnější informace viz práce *Aplikace pro efektivní řízení dronu s využitím rozšířené virtuality* (Hubinák, 2020, FIT VUT).

Mise jsou definované ve formátu JSON, konkrétně prostřednictvím souboru *DroCo_Data/StreamingAssets/mission.json*. Příklad velice základní mise je zobrazen na obrázku 5.

```
{
  "checkpoints": [
    {
      "drones": ["Drone1"],
      "type": "confirm",
      "name": "Point",
      "points": [
        {
          "height": 80,
          "latitude": 49.228634,
          "longitude": 16.597110,
        }
      ],
      "description": "Take a photo"
    }
  ],
  "drones": [
    {
      "name": "Drone1",
      "latitude": 49.226020,
      "longitude": 16.597149,
      "url":
    }
  ]
}
```

Obr. 5: Ukázka JSON zprávy popisující jednoduchou misi.

Server pro sdílení letových dat

Pro sdílení online letových dat všech reálných dronů, řízených přes aplikaci DroCo, slouží jednoduchý Python server⁵. Server funguje na modelu broadcast, kde přijímá zprávy formátu JSON (obsahující letová data) a následně rozesílá všem ostatním připojeným instancím aplikace DroCo. Připojení k serveru probíhá automaticky po startu aplikace, změně textového pole „DroCo Server IP“ nebo kliknutím na ikonku *Reconnect* v ovládacím panelu. Spojení se serverem je realizováno prostřednictvím technologie WebSocket. Zpráva obsahující letová data je tvořena osmi parametry:

- **DroneId** – unikátní, automaticky vygenerovaný identifikátor dronu
- **Altitude** – aktuální nadmořská výška dronu

⁵ https://github.com/robofit/drone_server

- **Latitude a Longitude** – aktuální zeměpisná šířka a délka
- **Pitch, Roll a Yaw** – aktuální rotace drona
- **Compass** – hodnota kompasu drona (pro rotaci)

Příklad zprávy lze vidět na obrázku 6.

```
{
  "DroneId": "English-5FF44E52-0-1F62EDA8",
  "Altitude": 239.7171630859375,
  "Latitude": 49.22674755347583,
  "Longitude": 16.596793361940916,
  "Pitch": 3.7097833156585695,
  "Roll": -0.6313909292221069,
  "Yaw": 90.0,
  "Compass": 286.5473327636719
}
```

Obr. 6: Ukázka JSON zprávy popisující letová data ovládaného dronu. Zprávy s aktuálními daty jsou posílány na server za účelem synchronizace všech kooperujících dronů.

Aktuální realizace serveru je navržena a implementována pouze pro experimentální účely. Řešení neobsahuje žádné bezpečnostní prvky, optimalizační postupy apod.

Uživatelský manuál

Kapitola popisuje průvod prvním spuštěním, představuje vyvinuté prvky pro usnadnění pilotování, snížení mentální zátěže a zvýšení orientace při plnění misí. Je rovněž popsáno jak aplikaci a jednotlivé prvky používat. Pro zprovoznění aplikace na zařízení Android stačí stáhnout APK soubor⁶ a nainstalovat jej (minimální podporovaná verze Android 7.0), pro zprovoznění na PC (Windows/Linux) jsou zveřejněny spustitelné binární soubory v publikovaném archivu⁷. Pro instalaci aplikace v prostředí Unity je dostupný návod popsany níže v kapitole Instalační manuál.

Základní nastavení

Při prvním spuštění aplikace mají všechny nastavitelné parametry předvolenou hodnotu. Pro jejich změnu je k dispozici panel s nastavením (viz obr. 7), který se uživateli zpřístupní po kliknutí na tlačítko v levém horním okraji obrazovky. Prvním parametrem, který je potřeba nastavit, je střed vygenerované mapy v poli „Map Center“. Velikost mapy je možné nastavit posuvníkem „Map Size“ a to od 1200x1200 metrů (úroveň 1) po 8400x8400 metrů (úroveň 10). Terén mapy je možné nastavit tlačítkem „Change Map Source“. Na výběr je satelitní snímek nebo klasická mapa. Pro stažení mapových podkladů je nutné internetové připojení. Po nastavení těchto parametrů je možné načíst misi stiskem tlačítka „Load Mission“. Misi může uživatel definovat v souboru *DroCo_Data/StreamingAssets/mission.json*.

Pro připojení k dronu je potřeba vyplnit pole „RosBridge IP“, obsahující url *rosbridge* serveru, který běží na dronu. Jelikož je spojení realizováno pomocí technologie WebSocket, je nutné vyplnit pole ve formátu *ws://ip:port*. Pro zobrazení živého obrazu z kamery dronu je potřeba vyplnit pole „Video Topic“ patřičným názvem „topicu“ (ROS terminologie pro datový kanál), který obsahuje video z kamery. U videa je potom možné nastavit jeho rozlišení a zorné pole kamery (FOV – field-of-view). Tlačítkem „Set Altitude Offset“ je možné ručně upravit posun výšky dronu pro případy, že měření nadmořské výšky dronu není přesné. Další parametry potom umožňují zapínat/vypínat některé funkce aplikace (virtuální plátno pro FPV zobrazení video-streamu z dronu, zobrazení budov, zobrazení navigačních šipek, uživatelem definovaných zón za běhu aplikace apod.)

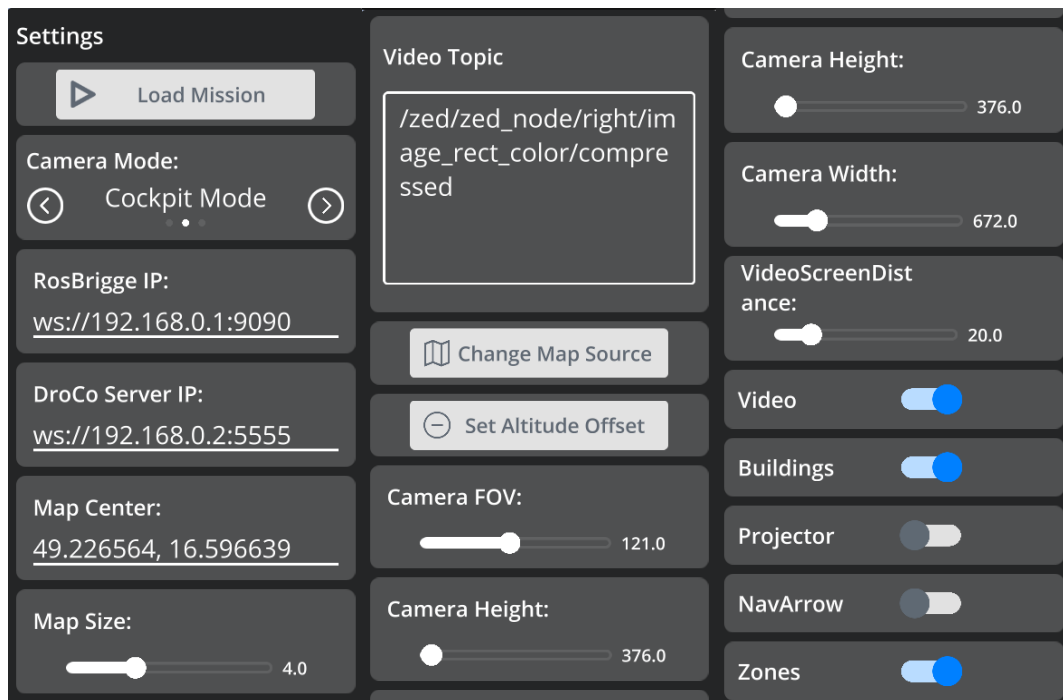
Pokud se dron bude podílet na více-dronové misi, tzn. bude potřeba sdílet data, je třeba nastavit IP adresu zařízení, na kterém je spuštěn server pro sdílení těchto letových dat (formát *ws://ip:port*). Po úpravě textového pole se rozhraní pokusí automaticky navázat spojení, o čemž uživatele indikuje výpis v horním panelu.

Na závěr má uživatel možnost si zvolit režim kamery (políčko *Camera Mode*). Podporovány jsou režimy:

- **Standard** – Kamera sleduje dron a zachovává si svou rotaci. Z uživatelova pohledu se tak naklápí pouze dron samotný a plátno s obrazem z kamery, scéna je statická.
- **Cockpit** – Kamera sleduje dron a kopíruje jeho rotaci. Z uživatelova pohledu se tak dron, společně s obrazem z kamery, jeví staticky, scéna a vše ostatní se naklápí.
- **Free** – Kamera je vyjmuta ze sledování dronu, lze s ní volně pohybovat po scéně.

⁶ https://github.com/robofit/drone_vstool/releases/download/0.1.0/DroCo.apk

⁷ https://github.com/robofit/drone_vstool/releases/tag/0.1.0



Obr. 7: Ukázka panelu s nastavením rozhraní. Nastavení lze měnit přímo za běhu s okamžitým efektem.

Rozhraní a uživatelské prvky

Uživateli je k dispozici grafické uživatelské rozhraní, využitelné jak pro pilota, tak pro velitele mise. Poskytuje několik prvků usnadňujících orientaci, snižujících mentální zátěž a zvyšujících komfort letu při plnění mise. Konkrétně se jedná o tyto prvky (viz obrázky 1. a 8.):

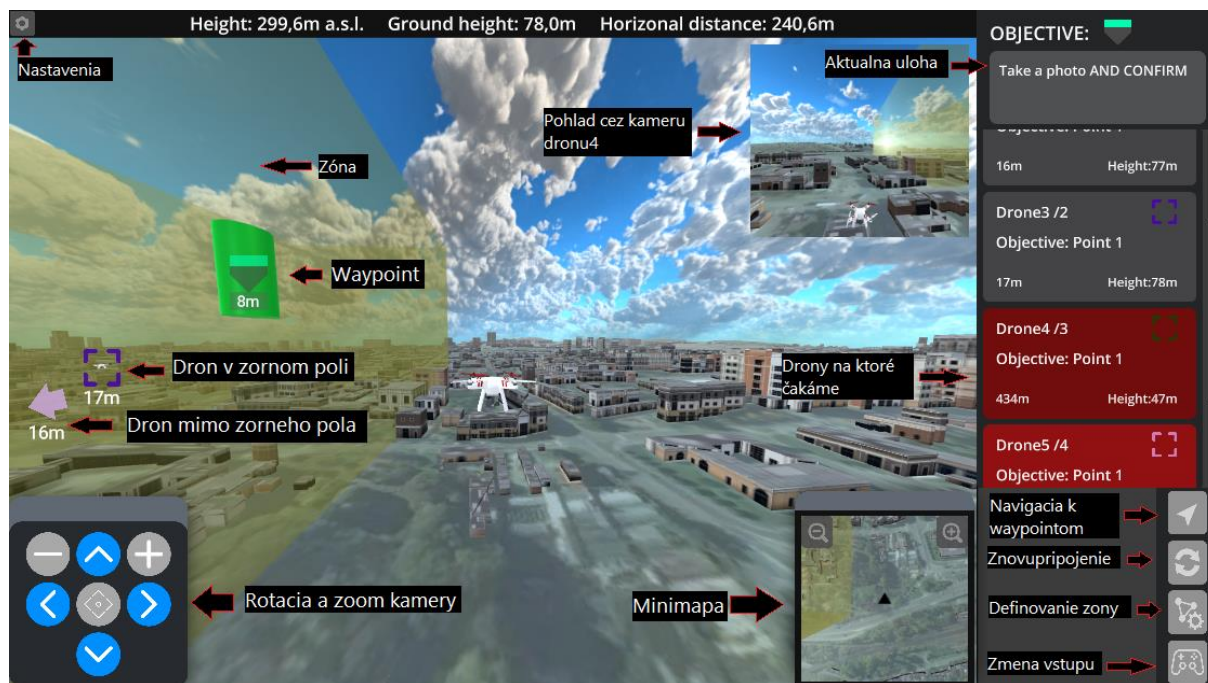
- grafická vizualizace mise: waypointy, zóny, úkoly, seznam dronů podílejících se na misi, náhled kamery z pohledu participujícího dronu,
- vizualizace ostatních dronů a dronů mimo zorné pole kamery (včetně vzdáleností od pilotovaného dronu),
- vizualizace vzdáleností a směrů k okolním překážkám,
- zobrazení letových dat (nadmořská výška, výška nad zemí, vzdálenost k místu vzletu),
- minimapa,
- panel pro ovládání rotace kamery.

Základní funkce, využitelné za běhu aplikace, jsou dostupné pomocí tlačítek v pravém dolním rohu obrazovky (viz obr. 8). Nejspodnější tlačítko slouží pro změnu zdroje letových dat. Dostupné jsou 3 zdroje: simulátor (ikona s gamepadem), živá data z dronu (ikona ROS) a náhodná data (ikona RAND). Další tlačítko slouží ke znovu-připojení ke dronu v případě výpadku nebo změny IP adresy RosBridge. Tlačítko se šipkou umožňuje uživateli definovat waypointy za běhu aplikace a to buď na místo, kde se dron nachází, nebo na místo kam uživatel klikne. Poslední tlačítko slouží k editaci zóny, kterou může uživatel definovat za běhu aplikace. Je možné umístit pouze jednu takovou zónu a její hranice je možné definovat kliknutím. Editace se vypne po opětovném stisknutí tohoto tlačítka.

Na rotaci kamery slouží skupina tlačítek v levém dolním rohu obrazovky (viz obr. 8). Polohu těchto tlačítek lze libovolně měnit tažením po obrazovce. V pravé straně obrazovky se nachází informační panel obsahující popis aktuální úlohy, seznam dronů podílejících se na misi a minimapu. Na prvky seznamu dronů lze kliknout, čímž se zobrazí pohled přes kameru zvoleného dronu. Pokud je prvek červený, znamená to, že konkrétní dron se podílí na stejné úloze jako uživatel a ještě se nedostal ke zvolenému waypointu. Minimapu lze přibližovat a oddalovat a rovněž lze její polohu měnit tažením po obrazovce. Rozložení jednotlivých prvků rozhraní je ukázáno na obrázku 8.

Při letu v blízkosti budov, jejichž vzdálenosti jsou analyzovány na základě offline 3D modelu prostředí, se uživateli pod dronem zobrazí osmiúhelník s jednotlivými vzdálenostmi k překážce v daném směru (viz obr. 1). Ukazatele vzdáleností k překážkám pod/nad dronem jsou zobrazeny v dolní/horní části obrazovky.

V závislosti na využití aplikace DroCo se liší její ovládání. Je-li připojena k reálnému dronu, uživatel využívá pro interakci dotyky na obrazovku/klikání myši (Android/Desktop). Je-li ovšem aplikace využita jako simulátor (vstup je přepnut na zdroj *simulátor* – ikona s gamepadem), uživatel využívá pro interakci, mimo nastavování parametrů prostřednictvím dotyků/myši, gamepad (zejména pro létání se simulovaným dronem a ovládání kamery). Lze teoreticky využít jakýkoliv gamepad (testovány byly Xbox One S Controller a Playstation Dualshock 3), jen je nutné ověřit správné nastavení vstupů v *Unity Input Manageru*⁸.



Obr. 8: Grafické uživatelské rozhraní aplikace DroCo (na obrázku se simulovaným dronem a misí, bez živých dat).

⁸ <https://docs.unity3d.com/Manual/class-InputManager.html>

Instalační manuál

Návod na zprovoznění projektu v prostředí Unity, včetně všech potřebných závislostí a knihoven, je popsán na github wiki stránce projektu⁹. Aplikace byla vyvíjena pod operačním systémem Windows 10. Z github repozitáře lze rovněž stáhnout hotový build aplikace (verze pro Windows, Linux a Android)¹⁰. Návod na zprovoznění serveru pro sdílení letových dat mezi jednotlivými klienty aplikace DroCo je popsán na github wiki stránce projektu¹¹. Pro připojení k serveru z aplikace DroCo pak jen stačí vyplnit pole „DroCo Server IP“ ve formátu: *ws://hostname:port*.

⁹ https://github.com/robofit/drone_vstool#installation

¹⁰ https://github.com/robofit/drone_vstool/releases/tag/0.1.0

¹¹ https://github.com/robofit/drone_server#drone_server

Reference

PLASCENCIA Alfredo Ch., BERAN Vítězslav, NOSKO Svetozár. *Octomap Representation for Advanced Drone Control for Augmented Reality*. Brno, 2020. Technical report. VRASSEO.

HUBINÁK Róbert. *Aplikace pro efektivní řízení dronu s využitím rozšířené virtuality*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. 2020-08-26. Vedoucí práce Beran Vítězslav. Dostupné z: <https://www.fit.vut.cz/study/thesis/22839>

SEDLMAJER Kamil, BAMBUŠEK Daniel a BERAN Vítězslav. Effective Remote Drone Control Using Augmented Virtuality. In: *Proceedings of the 3rd International Conference on Computer-Human Interaction Research and Applications 2019*. Vienna: SciTePress - Science and Technology Publications, 2019, s. 177-182. ISBN 978-989-758-376-6.

SEDLMAJER Kamil. *Uživatelské rozhraní pro řízení dronu s využitím rozšířené virtuality*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. 2019-06-14. Vedoucí práce Beran Vítězslav. Dostupné z: <https://www.fit.vut.cz/study/thesis/16730/>