

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Inštalačná príručka

Dokument k tímovému projektu



B_{EHA} W_{ORKS}

Ághová Adriana
Civáň Martin
Cukerová Zuzana
Kováč Matúš

Pilňan Matúš
Schmidt Andrej
Villár Viliam

Úvod

Dokument „inštalačná príručka“ predstavuje postupy, vďaka ktorým je možné vytvoriť, nainštalovať, prípadne nakonfigurovať prototyp vytvorený našim tímom Behawork. Dokument obsahuje okrem už samotného postupu pre spustenie prototypu aj pomocné postupy, ktoré bolo taktiež pri práci na finálnom produkte potrebné vykonať.

1. Logger

Doplňujúce informácie

- Základný logger na logovanie pohybu vo VR spolu so stavom ovládaču napísaný vo [Flask-u](#) s využitím [flask_restplus](#).
- Ako úložisko dát slúži MongoDB
- Vyžaduje sa Python 3 alebo novší. Na nainštalovanie všetkých závislostí z requirements.txt odporúčame použiť nástroj pip.

Spustenie

1. Skontrolovať závislosti, najmä Python 3 a pip, prípadne si vytvoriť a aktivovať Python virtual enviroment.
2. Pullnúť tento repozitár.
3. Inštaláciu vykonať príkazom: *pip install -e .* z koreňového priečinku tohto repozitára.
4. Nastaviť konfiguráciu v súbore *server/config/config.py* podľa predlohy *server/config/default.py* najmä pripojenie k MongoDB.

Primárnym spúšťačom je súbor *logger.py*, no do produkcie odporúčame použiť niektorý z produkčných Python serverov, napríklad Gunicorn.

Dokumentácia je k dispozícii na adrese `<URL_PREFIX>/documentation`. Štruktúra údajov v API je inšpirovaná údajmi z OpenVR.

Vizualizácie

Na adrese `/visualisations` je možné zobrazíť vizualizácie nalogovaných dát. Vizualizácie sú rozdelené podľa mena používateľa. Každý používateľ má niekoľko sessions. Jedna session reprezentuje jeden pohyb.

Na opätovné spustenie animácie grafu treba obnoviť stránku.

Spušťanie v Dockeri

Treba mať nainštalovaný Docker. Na Windowse je to komplikovanejšie, s W10 Home Edition to nefunguje, ale W10 Education je pre študentov FIIT zadarmo.

V konfigurácii musí byť URL nastavená na adresu stroja (172.*) na ktorom beží Docker z pohľadu kontajnera, localhost dá len adresu kontajnera. Adresa sa dá zistiť príkazom `hostname -I` v kontajneri.

Pri spúšťaní treba prepojiť lokálny súborový systém a porty s kontajnerom:

```
docker run -p 8000:8000 -v <C:/cesta/k/repozitaru>:/logger --name behapass-server behapass-server
```

<C:/cesta/k/repozitaru> treba nahradiť skutočnou cestou, mala by sa končiť `/logger`. Dvojbodka tam musí byť.

Pri vytváraní konfigurácie v Idei treba dať do Bind ports to čo je v prepínači `-p` a do Bind mounts to čo je vo `-v`.

Parametre docker imageu

Premenné prostredia (pri spustení):

- *port* - port na ktorom má počúvať aplikácia v docker containeri. Default: 8000. Tento port treba otvoriť pri spúšťaní kontajneru. Príklad na spustenie na inom porte (port 1234):

```
docker run -p 1234:1234 -v <C:/cesta/k/repositaru>:/logger --name behapass-server --env  
"port"="1234" behapass-server
```

- *db_host* a *db_port* - adresa a port, na ktorých počúva databáza. Default: 172.17.0.1 a 27017

Premenná pri builde: *copy* - čo všetko sa má skopírovať do imageu. Pri vývoji odporúčame použiť možnosť *requirements.txt* a prepojiť lokálne súbory cez `-v`. V produkcii netreba zadávať nič a defaultne sa nakopíruje všetko. Iné hodnoty spravia chyby.

2. Logger - Client

Časť inštaláčnej príručky Logger-Client predstavuje aplikáciu zaoberajúcu sa zbieraním údajov z VR ovládačov a ich následné odosielanie na server.

Inštalácia

Podporovaná verzia Python 3.5+.

Pre správne fungovanie treba mať nainštalované [SteamVR](#). Bez toho aplikácia len pošle ukážkové dáta na server (test API).

1. Naklonovať tento repozitár (`git clone https://bitbucket.org/behaworks/logger-client.git`).
2. V priečinku repozitára nainštalovať závislosti: `pip install -r requirements.txt`.
3. Do priečinka repozitára manuálne stiahnuť knižnicu [Triad OpenVR](#). Stačí stiahnuť súbor `triad_openvr.py`.
4. Pri prvej inštalácii treba vytvoriť súbor `config/config.json`, forma podľa `config/example_config.json` (stačí skopírovať a premenovať)

Použitie

1. Logger zistí počet pripojených ovládačov.
2. Logger čaká na vstup od používateľa.
3. Akonáhle používateľ stlačí tlačidlo nahrávania (nastaviteľné v konfigurácii), začne sa zaznamenávať pohyb. Pohyb sa zaznamenáva kým je dané tlačidlo stlačené.
4. Akonáhle používateľ pustí tlačidlo, nahrávanie skončí a zaznamenané údaje sa odošlú na server.

Pokiaľ sa nepodarí načítať VR rozhranie (môže byť nepripojený headset, prípadne vypnuté SteamVR), aplikácia iba načíta skôr nalogované dáta zo súboru a pokúsi sa odoslať ich na server. Toto slúži na otestovanie odosielania na server bez potreby mať pripojený VR headset.

Konfigurácia

Konfiguračný súbor `config/config.json` obsahuje nasledovné polia:

- `sid_length` - Dĺžka session ID stringu,
- `sample_rate` - Vzorkovacia frekvencia snímaného pohybu,
- `api_host` - URL, kde sa nachádza logovacie API,
- `button` - Tlačidlo, ktorým sa ovláda nahrávanie pohybu. Pokiaľ sa zadá neplatná hodnota, používa sa trigger,
- `button_options` - Nemeniť. Možnosti pre pole button (opisy pre ovládače HTC Vive):
 - `trigger` - spúšťač na spodnej strane ovládača, to čo sa dá spojito stláčať,
 - `ulButtonPressed` - stlačenie hociktorého tlačidla,
 - `ulButtonTouched` - dotyk hociktorého tlačidla (trigger a trackpad),
 - `trackpad_pressed` - stlačenie trackpadu (kruhová plocha na vrchu ovládača),
 - `trackpad_touched` - dotyk trackpadu,
 - `menu_button` - tlačidlo nad trackpadom,
 - `grip_button` - tlačidlo na boku.

3. Logger – Python – Client

Časť Logger-Python-Client predstavuje opis API Loggera. Jedná sa o automaticky generovaný Python balík. O automatické generovanie sa stará [Swagger Codegen](#):

- API verzia: 1.1
- Verzia balíku: 1.0.0
- Buildovací balík: io.swagger.codegen.languages.PythonClientCodegen

Požiadavky

Python 2.7 and 3.4+

Inštalácia

Pip inštalácia

Ak sa nachádza Python balík priamo na Githube, je možná inštalácia priamo odtiaľ:

```
pip install git+https://github.com/GIT_USER_ID/GIT_REPO_ID.git
```

(je možné, že budete musieť spustiť *pip* s root oprávnením:

```
sudo pip install git+https://github.com/GIT_USER_ID/GIT_REPO_ID.git )
```

Potom nainštalujte balík: *import behapass_client*

Nástroje na nastavenie

Inštalácia pomocou [Setuptools](#).

```
python setup.py install --user
```

(alebo *sudo python setup.py install* v prípade inštalácie balíka všetkým používateľom)

Potom nainštalujte balík: *import behapass_client*

Spustenie

Po vykonaní krokov opísaných v časti *Inštalácia*, pokračujte v spustení:

```
from __future__ import print_function  
import time  
import behapass_client  
from behapass_client.rest import ApiException  
from pprint import pprint
```

```
# create an instance of the API class
api_instance = behapass_client.LoggerApi(behapass_client.ApiClient(configuration))
x_fields = 'x_fields_example' # str | An optional fields mask (optional)

try:
    api_response = api_instance.get_button_record(x_fields=x_fields)
    pprint(api_response)
except ApiException as e:
    print("Exception when calling LoggerApi->get_button_record: %s\n" % e)
```

Dokumentácia koncových bodov API

Všetky URI sú viazané ku <https://localhost/api>

Trieda	Metóda	HTTP request
LoggerApi	get_button_record	GET /logger/buttons
LoggerApi	get_logger_record	GET /logger/
LoggerApi	get_movement_record	GET /logger/movements
LoggerApi	post_logger_record	POST /logger/
LoggerApi	post_lookup	POST /logger/lookup

Dokumentácia Modelov

- [ButtonRecord](#)
- [LoggerRecord](#)
- [LookupResult](#)
- [MovementRecord](#)

4. Logger Unity3D Client

Logger Unity3D predstavuje našu klientsku aplikáciu. Aplikácia predstavuje používateľské rozhranie pomocou, ktorého používateľ interaguje s nami implementovanou funkcionalitou. Základná logika aplikácie umožňuje používateľom sa registrovať a prihlásiť.

Inštalácia

Odporúčaná verzia Unity3D je **2019.3.2**, odkaz na **stiahnutie**:

- [Windows](#)
- [Mac](#)
- [Linux](#)

Spustenie

1. Stiahnite si tento repozitár.
2. Otvorte celý stiahnutý priečinok v Unity3D.
3. V záložke Project v priečinku Scenes otvorte scénu Main Menu.unity.