# Projekt: Drone Tracking – Shrnutí (aktualizovaná verze)

## Struktura projektu

■ src/ Zdrojové kódy  
 ■ main.py – Vstupní bod aplikace  
 ■ cameras.py – Správa vstupních zdrojů videa  
 ■ tracking.py – Logika detekce a sledování dronu  
 ■ gimbal.py – Řízení gimbalu (pan/tilt)  
 ■ logger.py – CSV logger pro ukládání výsledků  
 ■ legacy/track\_drone23.py – Původní monolitický skript  
  
■ tests/ Jednotkové testy (pytest)  
 ■ test\_legacy.py – Test původního skriptu  
 ■ test\_logger.py – Test loggeru  
 ■ test\_gimbal.py – Test gimbalu  
  
■ test\_videos/ Testovací videa

## Funkční části

- Původní skript `track\_drone23.py` spustitelný na testovacím videu.  
- Modulární verze (`main.py`) propojuje video, tracking, gimbal a logger.  
- CSVLogger ukládá výsledky do souboru.  
- Testy (`pytest`) ověřují funkčnost původního skriptu, loggeru a gimbalu.

## Role kamer

■ Kamera 1 – sledovací (širokoúhlá)  
 - Slouží k detekci a sledování pohybujícího se objektu (dron, pták, letadlo).  
 - Udržuje objekt v záběru a poskytuje souřadnice pro řízení gimbalu.  
  
■ Kamera 2 – záznamová (teleobjektiv)  
 - Slouží k detailnímu nahrávání sledovaného objektu.  
 - Směřuje na střed sledovaného objektu podle údajů ze sledovací kamery.

## Diagram modulů

main.py – Vstupní bod, řídí běh aplikace  
↓  
cameras.py – Zajišťuje vstup videa (kamera / soubor)  
↓  
tracking.py – Detekce a sledování objektů  
↓  
gimbal.py – Řízení pohybu gimbalu (pan/tilt)  
↓  
logger.py – Loguje data do CSV

## PowerShell příkazy

# Aktivace venv  
cd C:\Users\Milan\Projekty\Dron  
.\venv\Scripts\activate  
  
# Spuštění aplikace s testovacím videem  
cd src

python main.py --source test\_videos\drone1.mp4 --output runs/out.mp4 --show --pause\_frame 15 --winsize 800 --tracker MOSSE

python main.py --source test\_videos\drone1.mp4 --output runs/out.mp4 --show --pause\_frame 15 --winsize 800 --tracker MOSSE  
python main.py --source test\_videos\drone1.mp4 --output runs/out.mp4

OpenCV tracker (jednoduchý, výběr ROI) --python src/main.py --mode opencv --source test\_videos/drone1.mp4 --show --pause\_frame 20 --winsize 800

YOLO + DeepSORT (multi-tracking):--- python src/main.py --mode yolo --source test\_videos/drone1.mp4 --show --winsize 800  
  
# Spuštění testů  
cd ..  
pytest -v

## Známé problémy / TODO

- [ ] Propojení `gimbal.py` s reálným hardwarem (PWM/MAVLink).  
- [ ] Přidání podpory pro druhou kameru (teleobjektiv) pro detailní záznam.  
- [ ] Vylepšení sledovací logiky (např. DeepSORT, YOLO detekce).  
- [ ] Přidat více testovacích videí pro robustnější testy.  
- [ ] Možnost ukládání videa a logu synchronně.