**TEHNIČNA DOKUMENTACIJA – Carobotics**

Sestavni deli projekta:

* Hub (vozlišče med algoritmi in simulatorjem)
* AI (program za avtonomno vožnjo vozila)
* AirSim (vizualizacija parametrov vožnje)
* STM program za manualno kontrolo avtomobila

Programski jeziki:

* Python
* Golang
* C
* C#

Operacijski sistem: macOS Big Sur

1. **Hub**Programske zahteve:  
   - Python (verzija 3.7)  
   Knjižnjice**:**- airsim  
   - cv2  
   - numpy  
   - os  
   - base64  
   - json  
   - time  
   - flask  
   - decimal  
   - functools  
     
   Hub nam služi kot centralno vozlišče med algoritmi in AirSim, ki teče znotraj Unity.   
   Vozlišče je implementiramo z Flask framework-om in deluje kot server na katerega kličemo GET in POST zahteve.  
   Preko vozlišča lahko dobimo vse podatke iz simulatorja (hitrost, usmerjenost koles, pospešek, zaviranje, ročna zavora, slika iz kemere, lokacija avtomobila, podatki iz IMU senzorja, informacije o trku vozila, podatek kdo upravlja vozilo (algoritem, človek)).  
   Centralno vozlišče nam tudi omogoča kontrolo nad vozilom, kjer lahko sami nastavljamo parametre vožnje (pospeševanje, zaviranje, usmerjenost koles, ročna zavora).  
   Vozlišče poženemo z ukazom: *python3 hub.py*

1. **AI**

1. **AirSim**  
   Programske zahteve:  
   - Unity Hub  
   - Unity (verzija 2019.3.12f1)  
   - AirSim (verzija #TODO)

V podmapo UnityDemo znotraj mape projekta AirSim skopiramo našo mapo Assets v kateri so vsi objekti, ki se bodo pojavili v sceni in skripte (C#) za nadzor upodabljanja vizualizacije nad temi objekti. Nato v Unity uvozimo AirSim projekt in ga zaženemo. Po končanem nalaganju pristisnemo gumb Play in v začetnem meniju izberemo Car Mode. Unity nam naloži začetno sceno in lahko pričnemo z vožnjo.

1. **STM**Programske zahteve:  
   - Cube IDE (1.5.0)  
   - C  
   - Golang  
     
   Strojna oprema:  
   - STM F411E