

Vizualizačné nástroje pre

```
import cv2
import os
from multiprocessing import Process, Pipe
import random
import time
```

mobilitu v inteligentných mestských sídlach

```
import math
```

```
os.environ['SUMO_HOME'] = '/Users/Yzak/PycharmProjects/bakalarka/sumo-0.32.0'
if 'SUMO_HOME' in os.environ:
```

```
    toolPath = os.path.join(os.path.dirname(__file__), 'tools')
    sys.path.append(toolPath)
else:
```

sys myšlienka a význam. Vedomosti nadobudnuté počas štúdia danej problematiky budú využité

sys.path.append(os.path.join('Users', 'Yzak', 'PycharmProjects', 'bakalarka', 'sumoCnd', 'src', 'src/sumo-gui'))

pri tvorbe aplikácie pomocou nástroja na simuláciu dopravy v spojení s knižnicou programovacích

sumoBinary = "/Users/Yzak/PycharmProjects/bakalarka/sumo-0.32.0/src/sumo-gui"

sumoCnd funkcií zameranej predovšetkým na počítačové videnie v reálnom čase. V prvej kapitole

import tracI opisujeme znaky, charakteristiky a dimenzie inteligentného mesta. Druhá kapitola sa venuje

def tracI inteligentnej mobilita, tomu, prečo vznikajú aplikácie pre inteligentnú mobilitu a prečo je

tracI dôležitá. Tretia kapitola analyzuje technológie použité pri tvorbe aplikácie. Štvrtá kapitola

print("\n\tPID (simulator)....." + str(os.vehicles_left = 0

print("\n\tLEFT\t\tRIGHT\t\tREROUTED\tJAM_MAX\t\tSTATUS\n")

opisuje postup pri práci na modeli aplikácie pre inteligentnú mobilitu.

#traci.vehicle.subscribe("veh0", (tc.VAR_ROAD_ID, tc.VAR_LANEPOSITION))

change_left = []

change_left.append(0)

sum_Kľúčové slová: inteligentné mesto, inteligentná mobilita, SUMO simulátor, Simulator of

change_right = []

change_right.append(0)

Urban Mobility, OpenCV, vizualizačné nástroje, senzor na detekciu vozidiel

SUM_of_right = 0

rerouted = 0

jam_length = 0

max_jam = 0

Autor: Jic'hág Šoltés

max_jam1 = 0

Vedúci práce: doc. Ing. Jarmila Škrinárová, PhD

max_jam2 = 0

```
import os
from os import getpid
import sys
```

```
from time import sleep
from multiprocessing.connection import Client
from multiprocessing import Process, Pipe
```

```
def counter(pipe):
    print("\n\tOpenCV sensor.....")
```

print("\n\tPID (sensor).....")

video = 'video.avi'

capture frames from a video

cap = cv2.VideoCapture(video)

backsub = cv2.createBackgroundSubtractorMOG2()

isolate moving cars

Trained XML classifiers describes some features in the image

vehicles_left = 0

vehicles_change_left = []

trame_change_left.append(0)

vehicles_right = 0

frame_change_right.append(0)

frame_change_right.append(0)

frame_change_right.append(0)

output_p, input_p = pipe

uns if capturing has been initialized.

if cap.isOpened():

while True:

SUMO

Simulation of Urban Mobility

