Name: Vincze Nándor, first year computer science BSc student

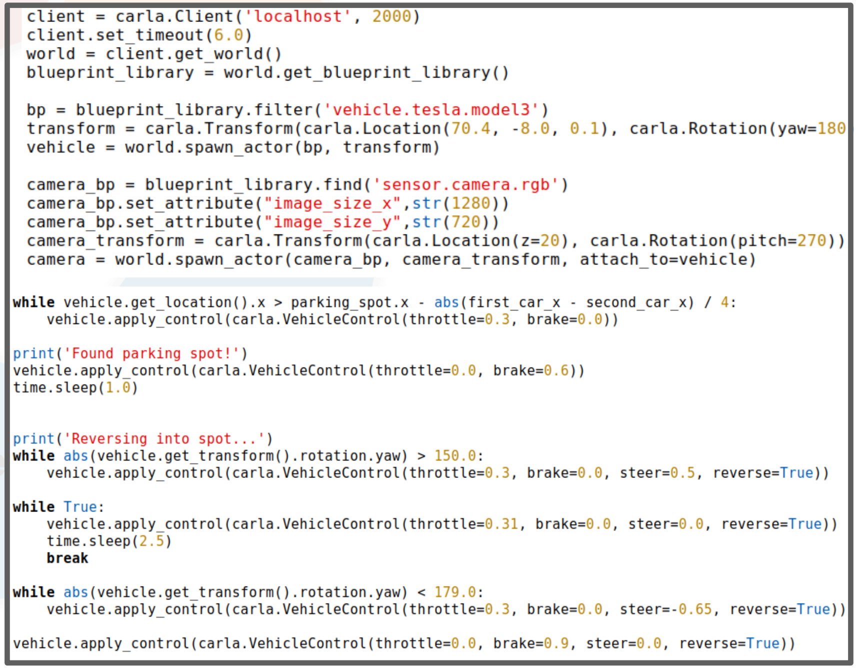
Project type: laboratory project

Topic: Vehicle Parking Parameter Optimalization

Supervisors: Dr. Balogh János, Dr. Bánhelyi Balázs

CARLA autonomous driving simulator

- Client-server architectre
- Python API
- Wide range of sensors (camera, radar, LiDAR, IMU…) and vehicles
- Dynamic weather and lighting
- Traffic scenario simulations, ML models, sensor data processing algorithms



```
 client = carla.Client('localhost', 2000)
 client.set_timeout(6.0)
 world = client.get_world()
 blueprint_library = world.get_blueprint_library()

 bp = blueprint_library.filter('vehicle.tesla.model3')
 transform = carla.Transform(carla.Location(70.4, -8.0, 0.1), carla.Rotation(yaw=180 ))
 vehicle = world.spawn_actor(bp, transform)

 camera_bp = blueprint_library.find('sensor.camera.rgb')
 camera_bp.set_attribute("image_size_x",str(1280))
 camera_bp.set_attribute("image_size_y",str(720))
 camera_transform = carla.Transform(carla.Location(z=20), carla.Rotation(pitch=270))
 camera = world.spawn_actor(camera_bp, camera_transform, attach_to=vehicle)

while vehicle.get_location().x > parking_spot.x - abs(first_car_x - second_car_x) / 4:
    vehicle.apply_control(carla.VehicleControl(throttle=0.3, brake=0.0))

print('Found parking spot!')
vehicle.apply_control(carla.VehicleControl(throttle=0.0, brake=0.6))
time.sleep(1.0)


print('Reversing into spot...')
while abs(vehicle.get_transform().rotation.yaw) > 150.0:
    vehicle.apply_control(carla.VehicleControl(throttle=0.3, brake=0.0, steer=0.5, reverse=True))

while True:
    vehicle.apply_control(carla.VehicleControl(throttle=0.31, brake=0.0, steer=0.0, reverse=True))
    time.sleep(2.5)
    break

while abs(vehicle.get_transform().rotation.yaw) < 179.0:
    vehicle.apply_control(carla.VehicleControl(throttle=0.3, brake=0.0, steer=-0.65, reverse=True))

vehicle.apply_control(carla.VehicleControl(throttle=0.0, brake=0.9, steer=0.0, reverse=True))
```

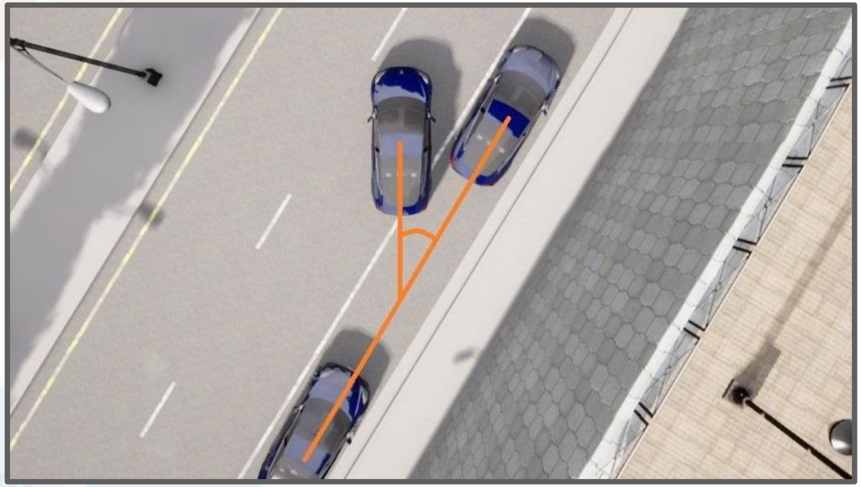
Internal

Experiments

Research
Diploma
Development

Reverse paralell parking parameters:

- Starting position (distance from center of parking spot):
  d_x
  d_y
- Angle of approach:
  λ
- Reversing into spot:
  v_r - velocity
  t_r - time
- Inverted steering angle:
  β_s



Internal

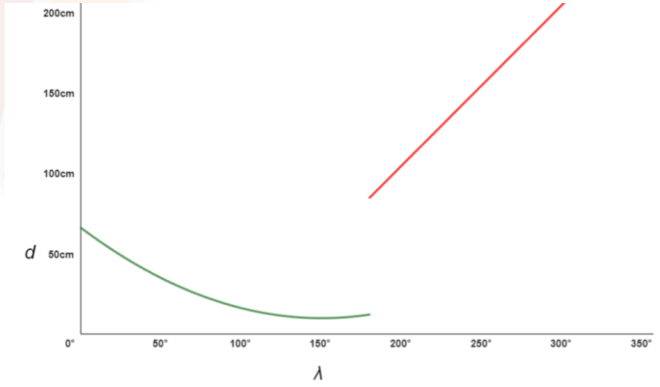
Results & future work

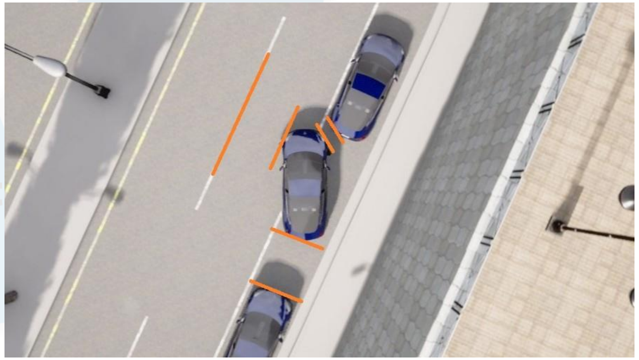
Description:

- Rule based reverse paralell parking based on geometric information

Future work:

- Penalty function based on collision position/distance form pavement
- Correction maneuvers
- Minimizing lane invasion/parking time
- Testing different vehicle dimensions, and turning capabilites
- Other pathplanning situations (overtaking, obstacle avoidance)





Internal