ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Δημιουργία Σεναρίων και Οδικών Χαρτών για τον Προσομοιωτή Οδήγησης CARLA με το Γραφικό Εργαλείο RoadRunner

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Κ. Κυριακίδης

Επιβλέπων: Παναγιώτης Τσανάκας

Καθηγητής Ε.Μ.Π.

Αθήνα, Απρίλιος 2024

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Δημιουργία Σεναρίων και Οδικών Χαρτών για τον Προσομοιωτή Οδήγησης CARLA με το Γραφικό Εργαλείο RoadRunner

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Κ. Κυριακίδης

Επιβλέπων: Παναγιώτης Τσανάκας

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 4η Απριλίου 2024.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
……	……	……
Παναγιώτης Τσανάκας Καθηγητής Ε.Μ.Π.	Ανδρέας-Γεώργιος Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Γεώργιος Ματσόπουλος Καθηγητής Ε.Μ.Π.

Αθήνα, Απρίλιος 2024

……

Δημήτριος Κ. Κυριακίδης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Τα συστήματα αυτόνομης οδήγησης (ADS) και υποβοήθησης οδηγού (ADAS) που ενσωματώνονται στα σύγχρονα αυτοκίνητα υποβάλλονται σε εικονικές δοκιμές προτού κριθούν υπό πραγματικές συνθήκες οδήγησης. Η εικονική δοκιμή πραγματοποιείται εκτελώντας σενάρια με την χρήση υπολογιστικών προσομοιώσεων, που περιλαμβάνουν οχήματα, περιβάλλον και κυκλοφοριακή ροή. Αυτή η μέθοδος συμβαδίζει με τις καθιερωμένες πρακτικές στον έλεγχο ηλεκτρονικών συστημάτων και έργων λογισμικού. Υποβάλλει το σύστημα οδήγησης σε συνήθεις αλλά και εξειδικευμένες κυκλοφοριακές καταστάσεις, προκειμένου να αξιολογήσει την αντίληψή του για το περιβάλλον και την ικανότητά του στην λήψη αποφάσεων, να επικυρώσει την ασφάλεια και να εντοπίσει τους περιορισμούς του. Η προσέγγιση αυτή αντιμετωπίζει δύο κύριες προκλήσεις: αφ' ενός την συγκέντρωση ενός εκτενούς καταλόγου τύπων σεναρίων που περιλαμβάνει όλες τις συνθήκες που ενδέχεται να αντιμετωπίσει το αυτόματο σύστημα υπό εξέταση και αφ' ετέρου την κλιμάκωση των δοκιμών με στόχο τον αξιόπιστο έλεγχο της ασφάλειας, ερευνώντας κρίσιμες περιπτώσεις των σεναρίων. Σημαντικό μέρος της έρευνας από την αυτοκινητοβιομηχανία επικεντρώνεται στην δεύτερη πρόκληση, ενώ έχουν γίνει σημαντικά βήματα προς την αντιμετώπιση της πρώτης. Τα πλέον διαδεδομένα πρότυπα αναπαράστασης οδικών δικτύων και σεναρίων σε προσομοιωτή είναι τα OpenDRIVE και OpenSCENARIO. Σκοπός της διπλωματικής εργασίας είναι η χρήση του γραφικού εργαλείου RoadRunner για την συγγραφή ολοκληρωμένων σεναρίων και στην συνέχεια η εκτέλεση αυτών στον προσομοιωτή οδήγησης CARLA. Μελετώνται και σχεδιάζονται στο RoadRunner δύο είδη σεναρίων βάσει συγκεκριμένων προδιαγραφών και κατασκευάζονται οι απαραίτητοι για αυτά χάρτες. Το πρώτο είδος αφορά δυνητικώς επικίνδυνο ελιγμό cut-in σε αυτοκινητόδρομο και το δεύτερο εξετάζει μία αυτοκινητοπομπή σε κυκλικό κόμβο. Επιπλέον αξιοποιείται η κατάλληλη προγραμματιστική διεπαφή του RoadRunner για την αυτοματοποίηση της παραγωγής παραλλαγών του σεναρίου cut-in. Ύστερα αναλύεται η διαδικασία μεταφοράς και εκτέλεσης των σεναρίων αυτών στον προσομοιωτή CARLA, τονίζοντας σε κάθε στάδιο τον βαθμό συμβατότητας μεταξύ των δύο εφαρμογών. Τελικώς αποτιμάται η επάρκεια του RoadRunner ως προς τον σχεδιασμό χαρτών και σεναρίων και του CARLA ως προς την δυνατότητα εκτέλεσης αυτών των παραχθέντων σεναρίων και διαπιστώνεται ότι ο συνδυασμός των εργαλείων εμφανίζει ικανοποιητικά αλλά όχι ιδανικά επίπεδα διαλειτουργικότητας.

Λέξεις Κλειδιά: Αυτόνομη Οδήγηση, Σενάρια, Προσομοιωτής Οδήγησης, RoadRunner, CARLA, OpenDRIVE, OpenSCENARIO

Abstract

The automated driving and driver assistance systems integrated into modern cars undergo virtual tests before being evaluated under real world driving conditions. Virtual testing entails scenario execution using computer simulations, typically encompassing vehicles, environment, and traffic. This method, known as scenario-based testing, mirrors established practices in the assessment of electronic systems and software. It involves subjecting the driving system to commonly encountered as well as specialized traffic scenarios, to evaluate its perception of the environment and decision-making ability, ascertain safety, and identify limitations. This approach presents two primary challenges: first, the compilation of a comprehensive list of scenario types including all pertinent situations that may be encountered by the automated system under examination; and second, the scaling of testing to achieve robust safety validation by exploring crucial instances of these scenarios. While ongoing research within the automotive industry tends to the latter challenge, significant progress has been made in addressing the former. The most common standards for representing road networks and scenarios in a simulator are OpenDRIVE and OpenSCENARIO. The purpose of this thesis is to use the graphical tool RoadRunner to design complete scenarios and then execute them in the CARLA driving simulator. Two types of scenarios are studied and designed in RoadRunner based on specific requirements, and the necessary maps are constructed for them. The first type involves a potentially hazardous cut-in maneuver on a highway, and the second examines a car convoy in a roundabout. Additionally, the appropriate programming interface of RoadRunner is utilized to automate the generation of variations for the cut-in scenario. Subsequently, the process of transferring and executing these scenarios in the CARLA simulator is analyzed, emphasizing the compatibility between the two applications at each step. Finally, the adequacy of RoadRunner for map and scenario design, as well as CARLA's capability of executing these generated scenarios are assessed, and it is concluded that the combination of RoadRunner and CARLA exhibits satisfactory but suboptimal levels of interoperability.

Keywords: Autonomous Driving, Scenarios, Driving Simulator, RoadRunner, CARLA, OpenDRIVE, OpenSCENARIO

Περιεχόμενα

Περίληψη **6**
Abstract **7**
Περιεχόμενα **8**
Ευρετήριο Εικόνων **10**
Ακρωνύμια **11**
Αντί Προλόγου **12**
Κεφάλαιο 1: Αντικείμενο της Έρευνας **14**
1.1 Εισαγωγή 14
1.2 Αυτόνομα οχήματα 14
1.2.1 Ορισμοί 14
1.2.2 Έρευνα και ανάπτυξη 15
1.2.3 Επίπεδα αυτονομίας 15
1.3 Προσομοιωτές 17
1.3.1 Ορισμός 17
1.3.2 Πλεονεκτήματα 18
1.3.3 Μειονεκτήματα 18
1.4 Η έννοια του οδηγικού σεναρίου 19
1.4.1 Ορισμός 19
1.4.2 Δοκιμές αυτόνομων συστημάτων οδήγησης 20
1.4.3 Τρόποι δημιουργίας 20
1.5 Σκοπός της έρευνας 21
1.5.1 Αντικείμενο 21
1.5.2 Σχετικές εργασίες 21
Κεφάλαιο 2: Βασικά Πρότυπα **22**
2.1 XML 22
2.2 ASAM 22
2.2.1 OpenDRIVE 22
2.2.2 OpenSCENARIO 25
Κεφάλαιο 3: Εργαλεία Ανάπτυξης **34**
3.1 Python 34
3.2 Unreal Engine 4 34
3.3 CARLA 35
3.3.1 Scenario runner 36

3.3.2 Server 36
3.3.3 Clients 37
3.3.4 Shell scripts 40
3.4 RoadRunner 41
3.5.1 Σχεδιασμός χάρτη 41
3.5.2 Συγγραφή σεναρίου 45
3.5 Υπολογιστικό σύστημα 48
3.6 Περιορισμοί 49
Κεφάλαιο 4: Υλοποίηση 51
4.1 Cut-in σε αυτοκινητόδρομο 51
4.1.1 Περιγραφή 51
4.1.2 Κατασκευή στο RoadRunner 52
4.1.3 Μεταφορά στον CARLA 54
4.1.4 Δημιουργία παραλλαγών 56
4.1.5 Συμπέρασμα 58
4.2 Αυτοκινητοπομπή σε κυκλικό κόμβο 59
4.2.1 Περιγραφή 59
4.2.2 Κατασκευή στο RoadRunner 60
4.2.3 Μεταφορά στον CARLA 62
4.2.4 Συμπέρασμα 66
Κεφάλαιο 5: Επίλογος 67
5.1 Πορίσματα 67
5.2 Μελλοντικές επεκτάσεις 67
Παράρτημα 68
Α': Σχεδιασμός του campus του Ε.Μ.Π. στο RoadRunner 68
Β': Εισαγωγή χάρτη από RoadRunner στον CARLA σε Windows 70
Βιβλιογραφία 73

Ευρετήριο Εικόνων

Εικόνα 1: Χαρακτηριστικό δείγμα ποιοτικής περιγραφής σεναρίου. 20
Εικόνα 2: Η reference line του δρόμου όπως προκύπτει από σύνδεση καμπυλών. 23
Εικόνα 3: Αναπαράσταση των πληροφοριών κωδικοποιημένων στο αρχείο OpenDRIVE. 24
Εικόνα 4: Κατηγορία λωρίδας που απαντάται σε σενάριο της εργασίας. 25
Εικόνα 5: Χαρακτηριστικό απόσπασμα αρχείου .xodr. 25
Εικόνα 6: Συνέργεια των προτύπων του ASAM στην σύνθεση ολοκληρωμένων σεναρίων. 26
Εικόνα 7: Η αρχιτεκτονική λογισμικού ενός προσομοιωτή. 26
Εικόνα 8: Οι πρώτες σειρές ενός αρχείου .xosc όπου μεταξύ άλλων ορίζεται ο χάρτης. 28
Εικόνα 9: Ετικέτα Vehicle και τα περιεχόμενα αυτής σε αρχείο .xosc. 28
Εικόνα 10: Το σύστημα συντεταγμένων και προσανατολισμού κατά ISO 8855 [26]. 29
Εικόνα 11: Global και Private Actions. 29
Εικόνα 12: Η αλληλεπίδραση μεταξύ οντοτήτων, δράσεων και ελεγκτών. 30
Εικόνα 13: Παράδειγμα συνθήκης εν σχέσει με άλλη οντότητα. 30
Εικόνα 14: Διάγραμμα αλληλεπίδρασης μεταξύ προσομοιωτή και scenario director. 31
Εικόνα 15: Απόσπασμα από ένα Act. 32
Εικόνα 16: Απόσπασμα από ετικέτα FollowTrajectoryAction και το περιεχόμενό της. 33
Εικόνα 17: Επιθεώρηση εισηγμένων γραφικών στον Unreal editor. 35
Εικόνα 18: Το παράθυρο του CarlaUE4. 37
Εικόνα 19: Το παράθυρο του PyGame client στο ίδιο στιγμιότυπο με τον server. 38
Εικόνα 20: Απεικόνιση ενός χάρτη σε κάτοψη και προοπτική. 42
Εικόνα 21: Σχεδίαση χάρτη με πρότυπο δορυφορική φωτογραφία. 43
Εικόνα 22: Δείγμα αρχείου .osm. 44
Εικόνα 23: Κόμβος της ΕΟ7 στο OpenStreetMap και αποτύπωση αυτού στο RoadRunner. 44
Εικόνα 24: Ο ίδιος χάρτης στο RoadRunner και στον CARLA εν μέσω προσομοίωσης. 45
Εικόνα 25: Κατάστρωση σεναρίου με τον logic editor και μεταβλητές. 47
Εικόνα 26: Η Town03 στον RoadRunner και μεταφορά σεναρίου στον CARLA. 48
Εικόνα 27: Σχηματική αναπαράσταση του cut-in [20]. 51
Εικόνα 28: Κάτοψη του χάρτη στον editor του RoadRunner. 53
Εικόνα 29: Οι δύο τύποι σεναρίου σε εκτέλεση στο RoadRunner. 54
Εικόνα 30: Στιγμιότυπα από την προσομοίωση. 56
Εικόνα 31: Η μέθοδος αλλαγής τιμών variables του RoadRunner. 57
Εικόνα 32: Ο κώδικας παραγωγής των παραλλαγών. 57
Εικόνα 33: Σχηματική αναπαράσταση του σεναρίου convoy. 59
Εικόνα 34: Δύο κατόψεις. Αριστερά από την κατασκευή του χάρτη, δεξιά η τελική μορφή. 61
Εικόνα 35: Προοπτική των δύο εκδοχών στο RoadRunner από το ίδιο σημείο θέασης. 61
Εικόνα 36: Χάραξη τροχιάς και καμπύλη χρόνου-θέσης για τα waypoints. 62
Εικόνα 37: Το επίμαχο τμήμα κώδικα της ChangeActorWaypoint. 64
Εικόνα 38: Διαδοχικά στιγμιότυπα από το σενάριο στους δύο χάρτες 65
Εικόνα 39: Κάτοψη του campus στο OpenStreetMap. 68
Εικόνα 40: Το επαγόμενο από το αρχείο .osm οδικό δίκτυο. 68
Εικόνα 41: Η τελική μορφή του χάρτη του campus στο RoadRunner. 69
Εικόνα 42: Εξαγωγή χάρτη από RoadRunner. 70
Εικόνα 43: Κατάλογος χαρτών στον Unreal Editor. 72

Ακρωνύμια

ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
ADS	Automated Driving System
AEB	Automatic Emergency Braking
ASAM	Association for Standardization of Automation and Measuring systems
ELK	Emergency Lane Keeping
EuroNCAP	European New Car Assessment Programme
GPS	Global Positioning System
ISO	International Organization for Standardization
LKAS	Lane Keeping Assist System
NHTSA	National Highway Traffic Safety Administration
SAE	Society of Automotive Engineers

Αντί Προλόγου

Η παρούσα διπλωματική εργασία εκπονήθηκε κατά το ακαδημαϊκό έτος 2023-2024 στο πλαίσιο των ερευνητικών δραστηριοτήτων του Ερευνητικού Πανεπιστημιακού Ινστιτούτου Συστημάτων Επικοινωνιών και Υπολογιστών που υπάγεται στην Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσοβίου Πολυτεχνείου.

Ευχαριστώ θερμά τον επιβλέποντα Καθηγητή μου κ. Παναγιώτη Τσανάκα για την ανάθεση της εργασίας και την ευκαιρία που μου έδωσε να ασχοληθώ με το τόσο ενδιαφέρον θέμα της αυτόνομης οδήγησης. Του εύχομαι ειλικρινά καλή δύναμη στον αγώνα του ως Κοσμήτωρ της Σχολής. Επίσης, ευχαριστώ τον Καθηγητή κ. Ανδρέα Σταφυλοπάτη και τον Καθηγητή κ. Γεώργιο Ματσόπουλο για την τιμή που μου έκαναν να συμμετάσχουν στην εξεταστική επιτροπή.

Σε όλα τα στάδια της εργασίας συνέδραμαν καθοριστικά οι ερευνητές της ομάδας I-Sense του ΕΠΙΣΕΥ Γεώργιος Χατζηπαυλής και Αναστασία Μπολοβίνου. Τους ευχαριστώ ιδιαιτέρως για την υπομονή και την προθυμία που έδειξαν να με βοηθήσουν.

Ανεκτίμητοι συμπαραστάτες σε αυτά τα χρόνια υπήρξαν βέβαια οι αγαπημένοι μου φίλοι και προπάντων οι γονείς μου.

Εἰπὲ δέ μοι γαῖάν τε τεὴν δῆμόν τε πόλιν τε,
ὄφρα σε τῇ πέμψωσι τιτυσκόμεναι φρεσὶ νῆες.
Οὐ γὰρ Φαιήκεσσι κυβερνητῆρες ἔασιν,
οὐδέ τι πηδάλι' ἐστί, τά τ' ἄλλαι νῆες ἔχουσιν·
ἀλλ' αὐταὶ ἴσασι νοήματα καὶ φρένας ἀνδρῶν…

ΟΔΥΣΣΕΙΑ

Κεφάλαιο 1: Αντικείμενο της Έρευνας

1.1 Εισαγωγή

Η αυτόνομη οδήγηση οχημάτων είναι ένα σχετικά νέο πεδίο εφαρμοσμένης έρευνας με σημαντικές επιπτώσεις στην βιομηχανία της αυτοκίνησης. Ήδη έχει τεθεί σε εφαρμογή σε διάφορες χώρες του κόσμου και αναμένεται να αλλάξει το τοπίο της μετακίνησης στις επόμενες δεκαετίες. Στην πρόσφατη ανάπτυξή της έχει συμβάλει τόσο η πρόοδος της τεχνητής νοημοσύνης όσο και η βελτίωση των αισθητήρων και συστημάτων πλοήγησης, υποστηριζόμενες φυσικά από την αύξηση της υπολογιστικής δύναμης των επεξεργαστικών μονάδων. Ύστερα από πολυετή πρόοδο στα συστήματα υποβοηθούμενης οδήγησης, οι μεγαλύτερες αυτοκινητοβιομηχανίες επενδύουν τώρα και στην τεχνολογία των αυτόνομων ή «αυτο-οδηγούμενων» (self-driving) οχημάτων, προσβλέποντας σε συστήματα οδήγησης όπου ο ανθρώπινος παράγοντας θα υπεισέρχεται όλο και λιγότερο.

Στο κεφάλαιο αυτό επιχειρείται η ανάλυση των θεμελιωδών εννοιών που απασχολούν την τεχνολογία αυτόνομων αυτοκινήτων και η επισκόπηση των σύγχρονων εξελίξεων και προβλημάτων αυτής, ώστε εν τέλει να διευκρινιστεί σαφώς ο σκοπός της εργασίας.

1.2 Αυτόνομα οχήματα

1.2.1 Ορισμοί

Οι Norvig και Russel [1] ορίζουν την αυτονομία ενός ευφυούς πράκτορα ως την ικανότητα αυτού να προσαρμόζεται βάσει των αισθητηρίων που προσλαμβάνει ώστε να συμπληρώνει ή να διορθώνει την γνώση με την οποία έχει αρχικώς εφοδιαστεί. Η ενσωμάτωση της μάθησης κατ' αυτόν τον τρόπο επιτρέπει τον σχεδιασμό ενός μόνο ορθολογικού πράκτορα ικανού να αντεπεξέλθει σε μεγάλη ποικιλία περιβαλλόντων.

Ο όρος «περιβάλλον εργασιών» στο πλαίσιο της μελέτης των ευφυών πρακτόρων συνοψίζει την σύνθεση των στοιχείων του προβλήματος: μέτρο απόδοσης (performance), φυσικό περιβάλλον (environment), μηχανισμοί δράσης (actuators) και αισθητήρες (sensors). Ειδικά στην περίπτωση της αυτόνομης οδήγησης, μέτρα απόδοσης είναι, μεταξύ άλλων, η μετάβαση στον σωστό προορισμό, η ελαχιστοποίηση της κατανάλωση καυσίμων και των φθορών, η ελαχιστοποίηση του χρόνου και κόστους του δρομολογίου, η αποφυγή τροχαίων παραβάσεων, και η μεγιστοποίηση της ασφάλειας των επιβατών και του οχήματος. Το περιβάλλον οδήγησης μπορεί να είναι μία ποικιλία δρόμων, από αγροτικούς χωματοδρόμους και επαρχιακές οδούς, έως πολύπλοκα οδικά δίκτυα με στενούς δρόμους και διασταυρώσεις εντός πόλεων, έως μεγάλες οδικές αρτηρίες πολλαπλών λωρίδων. Παράμετροι του περιβάλλοντος είναι επίσης η παρουσία άλλων οχημάτων και πεζών, οι κανόνες του οδικού δικτύου (όπως πλευρά οδήγησης στα αριστερά ή δεξιά), και οι καιρικές συνθήκες [1]. Οι μηχανισμοί οδήγησης που έχει στην διάθεσή του ένα αυτόματο όχημα είναι εν πολλοίς οι ίδιοι με αυτούς που έχει ο άνθρωπος οδηγός: έλεγχος του κινητήρα με το γκάζι, της κατεύθυνσης με το τιμόνι και έλεγχος των φρένων. Οι βασικοί αισθητήρες του οχήματος περιλαμβάνουν το ταχύμετρο, το οδόμετρο και τους αισθητήρες του κινητήρα και των μηχανικών ηλεκτρικών συστημάτων που ενημερώνουν για την κατάσταση του αυτοκινήτου. Εκτός αυτών, για την υποκατάσταση των ανθρωπίνων αισθήσεων, το αυτόνομο όχημα χρειάζεται συνδυασμό αισθητήρων όπως κάμερες ορατού

φάσματος (rgb), κάμερες υπερύθρου φωτός, GPS, radar (radio detection and ranging) και lidar (light detection and ranging) [2][3]. Η συσκευή lidar σαρώνει το περιβάλλον με μία ακτίνα φωτός, της οποίας η ανάκλαση δίνει πληροφορίες σχετικά με την θέση, την απόσταση και την ταχύτητα των αντικειμένων εν σχέσει με την ίδια.

Στο στάδιο της έρευνας βέβαια, ο βαθμός ενσωμάτωσης των ανωτέρω στοιχείων εξαρτάται από τις δυνατότητες και τους σκοπούς της ερευνητικής ομάδας. Είναι πολύ πιθανό, για παράδειγμα, να μελετηθούν οχήματα μόνο εντός αστικού δικτύου [3].

1.2.2 Έρευνα και ανάπτυξη

Το πρώτο ορόσημο στην ανάπτυξη της αυτόνομης οδήγησης είναι το 1989, όταν το σύστημα υπολογιστικής όρασης ALVINN εκπαιδεύτηκε να κατευθύνει ένα αυτοκίνητο ώστε να παραμένει σε μία λωρίδα. Τοποθετήθηκε στο εξοπλισμένο με βιντεοκάμερες και ελεγχόμενο από υπολογιστή όχημα NAVLAB του πανεπιστημίου Carnegie-Mellon, και το οδήγησε να διασχίσουν τις Ηνωμένες Πολιτείες, μία απόσταση 4500 χιλιομέτρων, κατά την οποία το αυτόματο σύστημα είχε τον έλεγχο στο 98% της διαδρομής, ενώ στο υπόλοιπο 2% έπρεπε να συνεπικουρήσει άνθρωπος [35]. Πιο πρόσφατα, το πεδίο γνώρισε μεγάλη ανάπτυξη κατά την δεκαετία του 2000 με τους διεθνείς διαγωνισμούς DARPA [41]. Πλέον όλα τα νέα παραγόμενα μοντέλα αυτοκινήτων διαθέτουν συστήματα ADAS (Advanced Driver Assistance Systems), τα οποία ενσωματώνουν μεταξύ πολλών άλλων αυτοματισμών ασφαλείας και τουλάχιστον κάποιες στοιχειώδεις λειτουργίες αυτόνομης οδήγησης (βλ. επόμενη ενότητα).

Ένα πλήρες σύστημα αυτόνομης οδήγησης (end-to-end driving system) αναμένεται να έχει τις εξής ικανότητες:

- Ανίχνευση αντικειμένων (object detection) βάσει των εισόδων που λαμβάνει από τους αισθητήρες του οχήματος,
- Αξιολόγηση του περιβάλλοντος και εκτίμηση συμπεριφοράς άλλων κινουμένων σωμάτων,
- Λήψη αποφάσεων, σχεδιασμός κίνησης και
- Έλεγχος του οχήματος.

Για την ανάπτυξη του απαραίτητου λογισμικού για τις ανωτέρω εργασίες χρησιμοποιούνται κυρίως νευρωνικά δίκτυα και ειδικότερα μέθοδοι βαθιάς μάθησης (deep learning) και ενισχυτικής μάθησης (reinforcement learning) [3].

1.2.3 Επίπεδα αυτονομίας

Η Ένωση Μηχανικών Αυτοκίνησης (Society of Automotive Engineers - SAE) [4][5] είναι διεθνής επαγγελματικός οργανισμός που ασχολείται με την προτυποποίηση και έκδοση προδιαγραφών σχετικά με αυτόνομα και μη επανδρωμένα οχήματα. Στο πεδίο των αυτοκινήτων διακρίνει έξι επίπεδα (levels) αυτονομίας, βάσει του ποσοστού ανάμειξης συστημάτων αυτόματης οδήγησης (Automated Driving System - ADS) στην οδηγική διαδικασία. Ως οδηγική διαδικασία (Dynamic Driving Task - DDT) ορίζεται από τον SAE το σύνολο όλων των λειτουργικών και στρατηγικών πράξεων που απαιτούνται σε πραγματικό

χρόνο για τον χειρισμό του αυτοκινήτου σε πραγματικές συνθήκες κίνησης. Τα επίπεδα αυτονομίας κατά τον SAE είναι:

- Επίπεδο 0: "No Driver Automation". Ο οδηγός έχει τον πλήρη έλεγχο και ευθύνη ανά πάσα στιγμή και το όχημα δεν διαθέτει κανένα αυτόματο σύστημα υποβοήθησης. Το όχημα σε αυτό το επίπεδο μπορεί βέβαια να διαθέτει σύγχρονα συστήματα ασφαλείας που όμως περιορίζονται στην προειδοποίηση του οδηγού για πιθανούς κινδύνους και στην στιγμιαία παρέμβαση σε περιπτώσεις εκτάκτου ανάγκης. Τέτοια είναι τα συστήματα Automatic Emergency Braking (AEB) και Anti-lock Braking System (ABS), και οι αισθητήρες τυφλού σημείου (blind spot warning) και παρέκκλισης από την λωρίδα (lane departure warning).

- Επίπεδο 1: "Driver Assistance". Το ADS είναι ικανό να πραγματοποιεί ένα μέρος της DDT, ελέγχοντας είτε την κατά μήκος (longitudinal) είτε την κατά πλάτος (lateral) κίνηση του αυτοκινήτου. Η μεν γίνεται με επιτάχυνση και πέδηση μέσω συστήματος adaptive cruise control (ACC), η δε με πλοήγηση μέσω συστήματος lane keep assist (LKAS). Ο οδηγός είναι υπεύθυνος για τα υπόλοιπα μέρη της οδήγησης σε κάθε περίπτωση και ανά πάσα στιγμή μπορεί να αναλάβει τον πλήρη έλεγχο.

- Επίπεδο 2: "Partial Driving Automation". Το ADS μπορεί να ελέγχει την κατά μήκος και την κατά πλάτος κίνηση ταυτοχρόνως. Το αυτοκίνητο, δηλαδή, διαθέτει και σύστημα ACC και LKAS. Συνεπώς, μπορεί να εξασφαλίσει σταθερή ταχύτητα και λωρίδα σε κάποια απλή διαδρομή σε αυτοκινητόδρομο, χωρίς να χρειάζεται ο οδηγός να κρατά το τιμόνι και να πατά γκάζι. Οι δυνατότητες και ευθύνες του οδηγού είναι οι ίδιες με του επιπέδου 1.

Μεταξύ του τρίτου και του τετάρτου επιπέδου γίνεται η μετάβαση από όχημα που διαθέτει αυτοματισμούς όμως ελέγχεται από τον άνθρωπο - οδηγό, σε όχημα αρκετά προηγμένο ως προς την αυτοματοποίηση ώστε να μπορεί να υποκαθιστά τον οδηγό. Στα τρία πρώτα επίπεδα κύριος υπεύθυνος για την αντίληψη του περιβάλλοντος και την αντίδραση σε γεγονότα (object and event detection and response) θεωρείται ο άνθρωπος, ενώ στα τρία τελευταία το ADS.

- Επίπεδο 3: "Conditional Driving Automation". Το ADS είναι ικανό να επιτελεί υπό τις κατάλληλες συνθήκες, παραδείγματος χάριν κυκλοφορία χαμηλής ταχύτητας σε συμφόρηση, ολόκληρη την οδηγική διαδικασία. Όμως, μπορεί να απαιτήσει από τον οδηγό να αναλάβει τον έλεγχο σε καταστάσεις που δεν μπορεί να χειριστεί, οπότε ο άνθρωπος οδηγός είναι ακόμη απαραίτητος και υπεύθυνος.

- Επίπεδο 4: "High Driving Automation". Ένα όχημα επιπέδου 4 είναι θεωρητικώς πλήρως αυτόνομο εντός του λειτουργικού του πεδίου (Operational Design Domain - ODD), ενώ εκτός αυτού μπορεί να απαιτεί έλεγχο από τον οδηγό. Το ODD περιορίζεται από παραμέτρους όπως η ταχύτητα, η κίνηση στον δρόμο, η επάρκεια φυσικού φωτός, και το ίδιο το οδικό δίκτυο. Χαρακτηριστικό παράδειγμα ενός οχήματος επιπέδου 4 θα ήταν ένα αυτοκίνητο ικανό να πλοηγηθεί εντελώς αυτόνομα εντός των ορίων ενός πανεπιστημιακού campus, μόνο την ημέρα και με χαμηλό όριο ταχύτητας. Όταν ενεργοποιείται το ADS, πρακτικά λειτουργεί ως μη-επανδρωμένο όχημα: αναιρείται η ιδιότητα του ανθρώπου οδηγού, και αυτός, αν υπάρχει, θεωρείται επιβάτης. Δεν είναι καν απαραίτητο το αυτοκίνητο να διαθέτει πεντάλ και τιμόνι, αλλά ακόμη και αν τα διαθέτει υπάρχει μία βασική διαφορά από όλα τα προηγούμενα επίπεδα: το ADS δεν υποχρεούται να παραδώσει

τον έλεγχο του οχήματος στον οδηγό ευθύς αμέσως μόλις αυτός τον απαιτήσει, αλλά μπορεί προτού ικανοποιήσει το αίτημά του να επιβεβαιώσει κάποιες συνθήκες ασφαλείας.

- Επίπεδο 5: "Full Driving Automation". Το όχημα έχει τις ίδιες δυνατότητες με το επίπεδο 4 αλλά το ODD του καλύπτει όλες τις δυνατές περιστάσεις. Είναι, δηλαδή, ένα πλήρως αυτόνομο όχημα υπό οποιεσδήποτε συνθήκες και σε οποιοδήποτε περιβάλλον.

Σύμφωνα με τα ανωτέρω, το ίδιο αυτοκίνητο μπορεί να φθάνει διαφορετικά επίπεδα αυτονομίας αναλόγως με το λειτουργικό πλαίσιο στο οποίο βρίσκεται.

Επίπεδα αυτονομίας στα σύγχρονα αυτοκίνητα

Πολλά αυτοκίνητα μαζικής παραγωγής (production cars) έχουν αυτονομία επιπέδου 2 ή και 3 σε αυτοκινητοδρόμους, και επιπέδου 4 ειδικά στην διαδικασία της στάθμευσης. Παραδείγματος χάριν, η Mercedes-Benz έχει εγκαταστήσει στα κορυφαία μοντέλα της (S-Class, EQE) συστήματα αυτονομίας επιπέδου 3 για οδήγηση σε αυτοκινητόδρομο με συμφόρηση με ταχύτητα έως 60km/h [6], ενώ για την μεμονωμένη εργασία της στάθμευσης έχει συστήματα επιπέδου 4 σε αρκετά μοντέλα [7]. Η BMW έχει επίσης αναπτύξει πολλές λειτουργίες συστημάτων επιπέδου 3, όπως τον "Highway Assistant" για οδήγηση χωρίς χέρια στο τιμόνι στον αυτοκινητόδρομο με ταχύτητα έως και 135km/h και τον "Lane Change Assistant" που πραγματοποιεί αλλαγή λωρίδας αυτόνομα ύστερα από έγκριση του οδηγού [8]. Πέραν των κορυφαίων τεχνολογικά μοντέλων και κατασκευαστών, πλέον τα περισσότερα νέα μοντέλα ενσωματώνουν συστήματα ασφαλείας με επίπεδο αυτονομίας 2, διαθέτοντας μεταξύ άλλων AEB, LKAS και ELK (Emergency Lane Keeping), ενώ ο εξοπλισμός με ABS είναι υποχρεωτικός διά νόμου στην Ευρωπαϊκή Ένωση. [9][10][11].

1.3 Προσομοιωτές

1.3.1 Ορισμός

Κατ' αρχάς ας αποσαφηνισθεί η διάκριση μεταξύ των εννοιών «προσομοίωση» και «εξομοίωση». Στο πεδίο της τεχνολογίας και οι δύο αναφέρονται γενικώς σε δοκιμές λογισμικού. Η μεν όμως προσομοίωση αναφέρεται ειδικότερα στην μοντελοποίηση και εικονική αναπαράσταση ενός πραγματικού φαινομένου (στην απλούστερη περίπτωση), ή ακόμη ακολουθίας γεγονότων εντεταγμένων σε περιβάλλον και των αλληλεπιδράσεων με αυτό. Η δε εξομοίωση αναφέρεται στην αναπαραγωγή με την μέγιστη δυνατή ακρίβεια της λειτουργίας μίας πραγματικής μηχανής ή διάταξης μηχανών. Τα υπολογιστικά συστήματα υλικού και λογισμικού που πραγματοποιούν αυτά τα έργα ονομάζονται αντιστοίχως προσομοιωτές και εξομοιωτές. Ο όρος «προσομοιωτής» θεωρείται μεταφραστικό δάνειο του αγγλικού "simulator", ενώ ο «εξομοιωτής» του "emulator".

Στην πράξη, τα συστήματα που αναπτύσσονται για την έρευνα στην αυτόνομη οδήγηση έχουν στον πυρήνα τους μία μηχανή φυσικής (physics engine), ένα λογισμικό δηλαδή που κωδικοποιεί τους νόμους του φυσικού κόσμου (κυρίως της μακροσκοπικής κλίμακος) στον ηλεκτρονικό υπολογιστή. Η μηχανή φυσικής μοντελοποιεί το υλικό περιβάλλον, τις κινήσεις των σωμάτων σε αυτό, τις κρούσεις με εμπόδια και άλλα σχετικά για την μελέτη της

αυτοκίνησης φαινόμενα, ωστόσο δεν αποσκοπεί στην ακριβή ψηφιακή αναπαραγωγή του κινητήρα και όλων των μηχανικών μερών του αυτοκινήτου. Συνεπώς, σύμφωνα με τους ανωτέρω ορισμούς, ένα τέτοιο σύστημα πρέπει να αποκαλείται μέσον προσομοίωσης.

Οι προσομοιωτές χρησιμοποιούνται κατά κόρον στην έρευνα και ανάπτυξη συστημάτων αυτόνομης οδήγησης, για τους λόγους που αναφέρονται στην συνέχεια. Εκτός από τον CARLA, άλλοι αναγνωρισμένοι προσομοιωτές είναι οι esmini, TORCS, SUMO και Assetto Corsa [3].

1.3.2 Πλεονεκτήματα

Η χρήση προσομοιωτών είναι ιδιαιτέρως ωφέλιμη στην διαδικασία ανάπτυξης και δοκιμασίας αυτομάτων συστημάτων. Εξοικονομεί χρόνο, καθώς επιτρέπει την εικονική εκτέλεση του ιδίου πειράματος ή της ιδίας δοκιμής επανειλημμένως και πολλές φορές σε σύντομο διάστημα. Επίσης μειώνει το κόστος έρευνας, αφού περισσότερα πειράματα στον προσομοιωτή συνεπάγονται λιγότερα χιλιόμετρα σε αληθινή κίνηση. Περιορίζεται, δηλαδή, το πλήθος των αναγκαίων αλλά δαπανηρών πραγματικών δοκιμών με οχήματα, που υπόκεινται στον κίνδυνο των υλικών ζημιών και της εγγενούς αβεβαιότητας του φυσικού περιβάλλοντος. Μάλιστα εκτιμάται ότι για την εγγύηση της ασφάλειας των ADS χωρίς την χρήση προσομοιωτών θα ήταν απαραίτητο να οδηγηθούν για 11 δισεκατομμύρια χιλιόμετρα [3]. Επιπλέον, προσομοιώνοντας τους διάφορους αισθητήρες του οχήματος (radar, lidar, cameras), μπορεί να δοκιμασθεί η απόκριση του συστήματος σε περιστάσεις που είναι δύσκολες στην αναπαραγωγή σε αληθινό πείραμα. Τέλος, ο προσομοιωτής παρέχει την δυνατότητα μελέτης της συμπεριφοράς του οχήματος σε ποικιλία μορφών οδικών δικτύων και σε διαφορετικές καιρικές συνθήκες.

1.3.3 Μειονεκτήματα

Το βασικό μειονέκτημα των προσομοιωτών είναι ότι δεν αναπαράγουν τέλεια όλες τις συνθήκες και απρόβλεπτες παραμέτρους του φυσικού περιβάλλοντος. Επιπλέον, δεν εγγυώνται πάντα την επαναληψιμότητα των πειραμάτων που εκτελούνται σε αυτούς, ειδικά όταν βασίζονται σε μηχανές γραφικών γενικού σκοπού, όχι προορισμένες ειδικά για προσομοιώσεις δοκιμών ασφαλείας.

1.4 Η έννοια του οδηγικού σεναρίου

1.4.1 Ορισμός

Η έννοια του σεναρίου έχει κεντρική θέση στην έρευνα τεχνολογίας αυτοκίνησης. Σύμφωνα με τον ορισμό του ASAM [12], σενάριο είναι μία περιγραφή της αλλαγής του περιβάλλοντος σε σχέση με τον χρόνο, από συγκεκριμένη προοπτική. Ο ορισμός αυτός ενέχει τόσο τα στατικά στοιχεία του περιβάλλοντος, δηλαδή το οδικό δίκτυο μαζί με τους κανόνες που το διέπουν και τα ακλόνητα αντικείμενα, όσο και τα δυναμικά στοιχεία, δηλαδή τα κινούμενα σώματα, τις καιρικές συνθήκες, τον φωτισμό και τις διάφορες μεταβαλλόμενες λογικές συνθήκες του οδικού δικτύου, όπως αυτές που προκύπτουν από φωτεινούς σηματοδότες. Ο ορισμός δεν διακρίνει μεταξύ σεναρίου που εκτυλίσσεται στον πραγματικό κόσμο ή σε υπολογιστική προσομοίωση.

Ο SAE ορίζει ως σενάριο την αφαιρετική και γενική περιγραφή μίας οδικής περίστασης ως προς τον χρόνο και τον χώρο, δίχως τον ακριβή καθορισμό των παραμέτρων. Θεωρεί περαιτέρω την έννοια του «λογικού σεναρίου» (logical scenario), ενός σεναρίου που θέτει αποδεκτά εύρη τιμών για τις παραμέτρους, και τέλος του «πραγματικού σεναρίου» (concrete scenario), μίας συγκεκριμένης περίπτωσης του λογικού με επακριβώς καθορισμένες τιμές για κάθε παράμετρο. Η διάκριση αυτή είναι ευρέως αποδεκτή από την σχετική βιβλιογραφία [13][14]. Σε κάθε περίπτωση, ο ορισμός διευκρινίζει ότι η συμπεριφορά του κεντρικού υπό μελέτη οχήματος δεν περιγράφεται από το ίδιο το σενάριο.

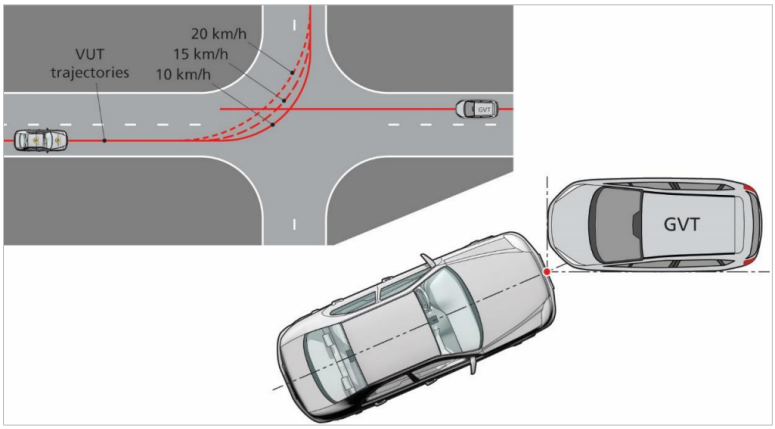
Ειδικά για τα συστήματα προσομοίωσης, ο ASAM θεωρεί ότι ένα πλήρες οδηγικό σενάριο αποτελείται αφ' ενός από την περιγραφή του στατικού περιβάλλοντος, η οποία περιλαμβάνει το λογικό οδικό δίκτυο και την γεωμετρική αποτύπωση του δρόμου και του περιβάλλοντος χώρου, και αφ' ετέρου την περιγραφή του «δυναμικού περιεχομένου», δηλαδή την καθολική σκηνοθεσία και συντονισμό της συμπεριφοράς των δυναμικών στοιχείων, και δυνητικώς και τα αλγοριθμικά μοντέλα στην οποία αυτή βασίζεται. Σε πρακτικό επίπεδο για τον σκοπό της εργασίας, η περιγραφή του οδικού δικτύου όταν αποδίδεται από τον προσομοιωτή ως μία τρισδιάστατη γραφική αναπαράσταση γεωμετρικών δεδομένων, είθισται να ονομάζεται «χάρτης».

Ο SAE ορίζει το όχημα "ego" (εκ του «εγώ») ως το επίκεντρο του σεναρίου, το όχημα που υπόκειται στις δοκιμασίες (tests) [4][5]. Αναλόγως με το σενάριο, αυτό μπορεί να σημαίνει ότι είναι το όχημα της προσομοίωσης που ελέγχεται από τον άνθρωπο - χρήστη, ή ότι είναι το όχημα που ελέγχεται από το υπό εξέταση αυτόνομο σύστημα. Στον κόσμο του RoadRunner όπου όλα τα οχήματα ελέγχονται από τον υπολογιστή, το ego έχει μόνο την δεύτερη σημασία. Ένα σενάριο μπορεί να περιέχει περισσότερα του ενός ego vehicles, ή και κανένα.

Επιπλέον, "Lead vehicle" ονομάζεται το όχημα που προπορεύεται του ego στην ίδια λωρίδα του δρόμου και "Reference vehicle" ονομάζεται ένα όχημα που τηρείται ως σημείο αναφοράς για τον καθορισμό της συμπεριφοράς άλλων οντοτήτων του σεναρίου. Η αναφορά μπορεί να είναι ως προς την θέση του, τα φυσικά μεγέθη της κινητικής του κατάστασης ή την συμπεριφορά του [19].

1.4.2 Δοκιμές αυτόνομων συστημάτων οδήγησης

Η χρήση των σεναρίων είναι αναγκαία για τον έλεγχο (testing), την επικύρωση και την πιστοποίηση ασφαλείας συστημάτων υποβοήθησης οδηγού (driver assistance) και συστημάτων αυτόνομης οδήγησης [15]. Η αυτοκινητοβιομηχανία, οι οργανισμοί πιστοποίησης και κρατικοί φορείς ορίζουν από κοινού καταλόγους πολλών διαφορετικών ειδών σεναρίων για την επίτευξη αυτών των στόχων. Ενδεικτικά, ο διεθνώς αναγνωρισμένος ευρωπαϊκός οργανισμός διενέργειας crash test EuroNCAP (European New Car Assessment Programme) κατηγοριοποιεί τις δοκιμασίες στις οποίες υποβάλλονται τα οχήματα καθορίζοντας σενάρια [16][17][18]. Η κατασκευή αυτών συμμορφώνεται με τον ανωτέρω ορισμό, δηλαδή περιλαμβάνει περιγραφή του οδικού δικτύου και ακριβή προσδιορισμό της κίνησης των οχημάτων προς κρούση. Όμοια μέθοδο ακολουθεί και ο αντίστοιχος οργανισμός των ΗΠΑ, ο NHTSA (National Highway Traffic Safety Administration) [23].



Εικόνα 1: Χαρακτηριστικό δείγμα ποιοτικής περιγραφής σεναρίου από τις προδιαγραφές του EuroNCAP.

Στην εικόνα 1 φαίνεται χαρακτηριστικό παράδειγμα σεναρίου ο EuroNCAP, προερχόμενο από το πρωτόκολλο AEB Car-to-Car Test Protocol [18]. Το συγκεκριμένο ονομάζεται “Car-to-Car Front turn-across-path” και απεικονίζει πλαγιομετωπική σύγκρουση. Η συνολική αξιολόγηση της ασφάλειας ενός ADS γίνεται εξετάζοντάς το σε πολλά διαφορετικά κρίσιμα σενάρια. Για κάθε ένα τέτοιο σενάριο, το πρωτόκολλο διευκρινίζει σαφώς τόσο την τοπολογία του οδικού δικτύου, όσο και τις ταχύτητες των οχημάτων κατά μήκος των τροχιών που ακολουθούν.

1.4.3 Τρόποι δημιουργίας

Οι Schütt et al. [14] διακρίνουν τρεις βασικές διαφορετικές μεθόδους πρόσκτησης σεναρίων:

- Παραγωγή εκ του μηδενός, συνήθως από ειδικούς που καθορίζουν προδιαγραφές όπως στα πρωτόκολλα ασφαλείας. Προκύπτουν σενάρια υψηλού επιπέδου αφαίρεσης (λογικά).
- Τροποποίηση ήδη υπάρχοντος σεναρίου για πειραματισμό με τις παραμέτρους. Η διαδικασία αυτή γίνεται και από αλγορίθμους για εντοπισμό κρίσιμων τιμών παραμέτρων πέραν των οποίων αλλάζει η τροπή των γεγονότων στο σενάριο.
- Εξαγωγή σεναρίου από αληθινά δεδομένα, όπως βίντεο κανονικής κυκλοφορίας σε έναν κόμβο. Προκύπτουν εξαιρετικά αληθοφανή concrete σενάρια.

1.5 Σκοπός της έρευνας

1.5.1 Αντικείμενο

Οι δοκιμασίες βάσει σεναρίων έχουν κεντρικό ρόλο στην διαδικασία επικύρωσης του ADS, ως μέθοδος περιγραφής καταστάσεων και δοκιμασίας του ως προς την αντίληψη του περιβάλλοντος και την λήψη ορθών αποφάσεων. Σε πρώτο στάδιο τα σενάρια διεξάγονται σε περιβάλλον υπολογιστικής προσομοίωσης. Η περιγραφή των σεναρίων από όλους τους αρμόδιους οργανισμούς γίνεται κατ' αρχήν σε αφηρημένο επίπεδο, με ποιοτική περιγραφή και σχηματική αναπαράσταση. Επομένως χρειάζεται βάσει αυτών των περιγραφών να μπορεί να αποτυπωθεί το σενάριο στον εκάστοτε προσομοιωτή.

Ο σκοπός της εργασίας είναι η χρήση του γραφικού σχεδιαστικού εργαλείου RoadRunner για την συγγραφή σεναρίων βάσει συγκεκριμένων προτύπων, και στην συνέχεια η μεταφορά ολόκληρων των σεναρίων αυτών στον προσομοιωτή CARLA. Το όφελος που αναμένεται από αυτήν την διαδικασία είναι η διευκόλυνση και επιτάχυνση του έργου κατασκευής σεναρίων για τον CARLA, καθώς και η αυτοματοποίηση της διαδικασίας παραγωγής παραλλαγών ενός λογικού σεναρίου.

Συγκεκριμένα στόχοι της έρευνας είναι:

- Ο σχεδιασμός χαρτών και σεναρίων στο RoadRunner. Η περιγραφή των κινήσεων των οχημάτων στα σενάρια είτε προσδιορίζοντας τα δυναμικά μεγέθη είτε χαράσσοντας τροχιά.
- Ειδικά για το σενάριο cut-in που περιγράφεται στο πρότυπο ISO 34502 [20] (βλ. Κεφάλαιο 4), η αυτοματοποίηση της παραγωγής παραλλαγών σεναρίου.
- Η μεταφορά χαρτών και σεναρίων από το RoadRunner στον CARLA μέσω των κοινών προτύπων κωδικοποίησης OpenDRIVE και OpenSCENARIO.
- Η εκτέλεση των σεναρίων στον προσομοιωτή.
- Η αξιολόγηση του RoadRunner ως προς το εύρος δυνατοτήτων συγγραφής σεναρίων και του CARLA ως προς την υποστήριξη των σεναρίων αυτών.

1.5.2 Σχετικές εργασίες

Υπάρχει αρκετή βιβλιογραφία σχετικά με το θέμα συγγραφής σεναρίων για προσομοιωτή. Ενδεικτικά, στις εργασίες των Gustafsson [21] και Heuer [15] επιχειρείται αυτοματοποιημένη παραγωγή κρίσιμων σεναρίων (critical scenarios) με χρήση OpenSCENARIO, για την δοκιμασία του ADS υπό επικίνδυνες συνθήκες. Οι Worrall et al. [13] χρησιμοποιούν αληθινά δεδομένα από lidar point cloud για την παραγωγή αληθοφανούς σεναρίου cut-in και κωδικοποιούν τα αποτελέσματα σε OpenSCENARIO. Ως προς τον σχεδιασμό τροχιάς, οι Takeda et al. [3] αναφέρουν και συγκρίνουν διάφορες μεθόδους: αλγορίθμους εύρεσης διαδρομής, περιγραφή με μαθηματικές καμπύλες, αριθμητική βελτιστοποίηση ή επεξεργασία πραγματικών δεδομένων (όπως συλλογή μετρήσεων lidar) από νευρωνικά δίκτυα. Από το ΕΠΙΣΕΥ έχουν εκπονηθεί και άλλες διπλωματικές εργασίες με σχετικά θέματα όπως η [51]. Τέλος, οι Sadeghi et al. [22] υλοποιούν στον προσομοιωτή CARLA ένα σενάριο cut-out, όπου το ego vehicle διαθέτει ACC και AEB και ακολουθεί προπορευόμενο όχημα, όταν το δεύτερο αλλάζει απότομα λωρίδα αποκαλύπτοντας εμπόδιο οπότε το ego καλείται να αντιδράσει εγκαίρως.

Κεφάλαιο 2: Βασικά Πρότυπα

2.1 XML

Η XML (“eXtensible Markup Language”) είναι μία γλώσσα σήμανσης, δηλαδή μία αναπαράσταση κειμένου που φέρει πληροφορία σε μορφή επισημάνσεων και ακολουθεί συγκεκριμένους κανόνες συντακτικής δομής. Μπορεί να χρησιμοποιηθεί τόσο για την αναπαράσταση δεδομένων, όσο και για την περιγραφή άλλων γλωσσών σήμανσης [47].

Οι δομικές μονάδες του αρχείου XML είναι οι ετικέτες (tags). Αυτές είναι διατεταγμένες ιεραρχικά σε δενδρική δομή, και περικλείουν τα δεδομένα. Κάθε ετικέτα μπορεί να περιέχει ιδιότητες (attributes), στις οποίες ανατίθενται τιμές. Επειδή η XML υποστηρίζει πλήρως το πρότυπο κωδικοποίησης χαρακτήρων Unicode, και προβλέπει την χρήση φυσικής γλώσσας για την ονοματοδοσία των ετικετών και των επιμέρους δεδομένων, τα αρχεία XML, παρ' ότι προορίζονται πρωτίστως για προγράμματα, είναι ευανάγνωστα και από τον άνθρωπο.

Το σημαντικότερο πεδίο εφαρμογής της XML είναι ως πρότυπο αναπαράστασης δεδομένων, για την επικοινωνία μεταξύ διαφορετικών υπηρεσιών. Σε αυτόν τον τομέα χρησιμοποιείται σε εφαρμογές διαδικτύου, αλλά και γενικώς σε εφαρμογές που απαιτούν αποθήκευση δεδομένων και ενσωματωμένο χειρισμό αυτών. Το καθοριστικό γνώρισμα της XML στο οποίο οφείλεται η διάδοσή της, είναι η επεκτασιμότητα (extensibility). Χάρη σε αυτήν, ο οιοσδήποτε χρήστης έχει την δυνατότητα να ορίσει ετικέτες και δομές δεδομένων σύμφωνα με τις ανάγκες του συστήματος που υλοποιεί. Σε αυτήν την δυνατότητα στηρίζονται τα πρότυπα ASAM που περιγράφονται στην συνέχεια, και τα δεδομένα OpenStreetMap.

2.2 ASAM

Ο οργανισμός ASAM e.V. (Association for Standardization of Automation and Measuring systems, σε ελεύθερη μετάφραση Ένωση για την Τυποποίηση συστημάτων Αυτοματισμού και Μέτρησης) ιδρύθηκε το 1998 από σύμπραξη σχεδόν όλων των γερμανικών κατασκευαστών αυτοκινήτων (Audi, BMW, Daimler, όμιλος VW). Έκτοτε έχουν λάβει μέρος σε αυτόν και άλλοι μεγάλοι κατασκευαστές (Denso, Ford, Renault) όπως επίσης και ο Διεθνής Οργανισμός Τυποποίησης ISO. Σκοπός του ASAM είναι η έκδοση προτύπων που αφορούν διάφορες τεχνολογίες αυτοκίνησης, όπως διαγνωστικές μετρήσεις, ανάπτυξη λογισμικού και αυτοματισμοί δοκιμών [46]. Συγκεκριμένα, στο πλαίσιο αυτής της εργασίας μελετήθηκαν τα πρότυπα που αφορούν την προσομοίωση, καθώς επί αυτών στηρίζονται τα προγράμματα συγγραφής σεναρίων και προσομοίωσης που χρησιμοποιήθηκαν στο πρακτικό μέρος.

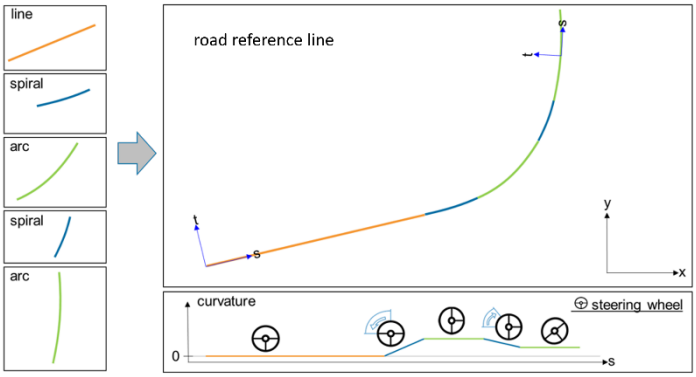
2.2.1 OpenDRIVE

Το πρότυπο ASAM OpenDRIVE καθορίζει ένα μοντέλο περιγραφής στατικών οδικών δικτύων, βασισμένο στην XML [24]. Μέρη του στατικού οδικού δικτύου μπορούν να είναι δρόμοι, λωρίδες, πεζοδρόμια, πινακίδες, διασταυρώσεις, νησίδες και άλλα αντικείμενα, όχι όμως κινούμενα σώματα. Την στιγμή συγγραφής της παρούσης εργασίας, η τελευταία έκδοση του προτύπου είναι η 1.8.0. Για την τοποθέτηση των διαφόρων στοιχείων του δικτύου στον χώρο, το πρότυπο χρησιμοποιεί τις γεωγραφικές συντεταγμένες όπως αυτές καθορίζονται από

το ISO 19111:2019 [25]. Η ορολογία που υιοθετεί για τα μηχανικά μεγέθη των οχημάτων υπαγορεύεται από το ISO 8855:2011 [26].

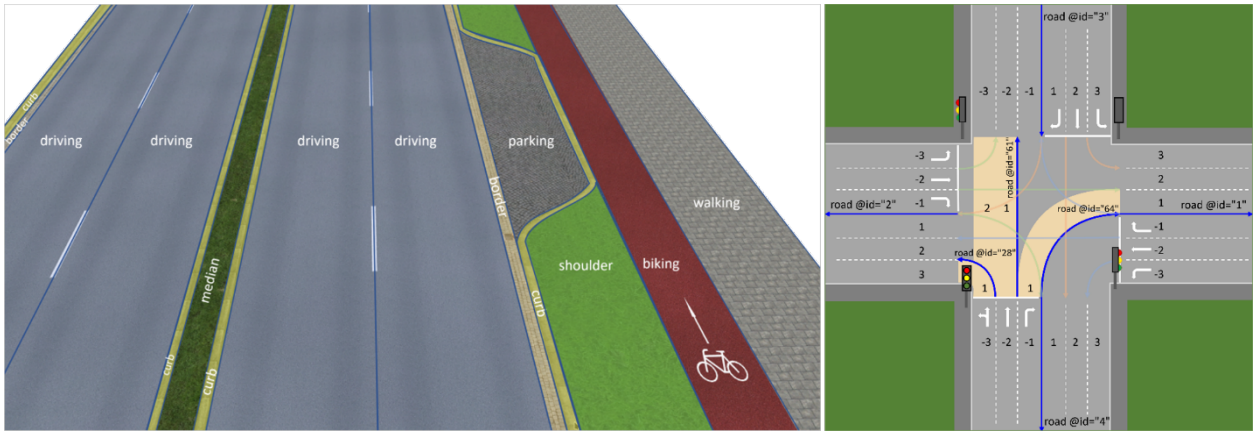
Δρόμοι

Η θεμελιώδης δομική μονάδα του οδικού δικτύου, ο δρόμος, περιγράφεται στο πρότυπο από μία γραμμή αναφοράς (reference line), τις επιμέρους λωρίδες που τον συνθέτουν, και συμπληρωματικά στοιχεία όπως πινακίδες και διαγραμμίσεις επί της ασφάλτου. Η γραμμή αναφοράς, η οποία καθορίζει το σχήμα του δρόμου, είναι γεωμετρική οντότητα συντιθέμενη από επιμέρους μαθηματικές καμπύλες, οι οποίες συνδυαζόμενες παράγουν συνεχές σχήμα. Το ύψος του δρόμου μπορεί να αναπαρασταθεί μέσω του προτύπου, καθορίζοντας σημεία ανύψωσης στην κάθετη τομή αυτού. Ένα σημαντικό στοιχείο του δρόμου που δεν περιγράφεται από το πρότυπο OpenDRIVE είναι η κατάσταση της επιφάνειας του δρόμου. Η πληροφορία σχετικά με το οδόστρωμα, όπως η ποιότητά του και κατ' επέκτασιν η πρόσφυση του δρόμου, και η γεωμετρική μορφολογία της επιφάνειάς του, απαραίτητη για την περιγραφή εξογκωμάτων και κοιλωμάτων, ενσωματώνεται από διακριτό πρότυπο του ASAM, το OpenCRG [24].



Εικόνα 2: Η reference line του δρόμου όπως προκύπτει από σύνδεση καμπυλών.

Όλα τα στοιχεία του δικτύου φέρουν ουσιώδη σημασιολογικά δεδομένα (semantic data) που τα αφορούν. Μία λωρίδα ενός δρόμου φερ' ειπείν εμπεριέχει, όπως φαίνεται εντός της ετικέτας `<label>`, πληροφορίες για το είδος της λωρίδας, το πλάτος αυτής, και εντός της υπο-ετικέτας `<speed>` το όριο ταχύτητας και την μονάδα μέτρησης στην οποία αυτό εκφράζεται, και άλλα σχετικά στοιχεία. Το πρότυπο προβλέπει σύνολο προκαθορισμένων κατηγοριών λωρίδας, που περιλαμβάνει μεταξύ άλλων λωρίδα αυτοκινήτων, ποδηλάτων, εισόδου σε αυτοκινητόδρομο και πεζών. Η τρέχουσα έκδοση του προτύπου δεν περιλαμβάνει κατηγορία για λωρίδα εκτάκτου ανάγκης (ΛΕΑ), ωστόσο αυτή η έλλειψη καλύπτεται εν μέρει από την υπό-ετικέτα `<rule>`, η οποία μπορεί να περιέχει ελεύθερο κείμενο. Ακόμη μία άξια αναφοράς υπο-ετικέτα της `<lane>` είναι η `<laneOffset>`, μέσω της οποίας επισημαίνεται η πλευρική μετατόπιση της αντίστοιχης λωρίδας από την γραμμή αναφοράς. Μέσω αυτής υλοποιούνται περισσότερο σύνθετες διατάξεις λωρίδων σε έναν δρόμο, όπου το πλήθος των λωρίδων ή η κατεύθυνση αυτών μπορεί να μεταβάλλονται κατά μήκος του δρόμου [24].



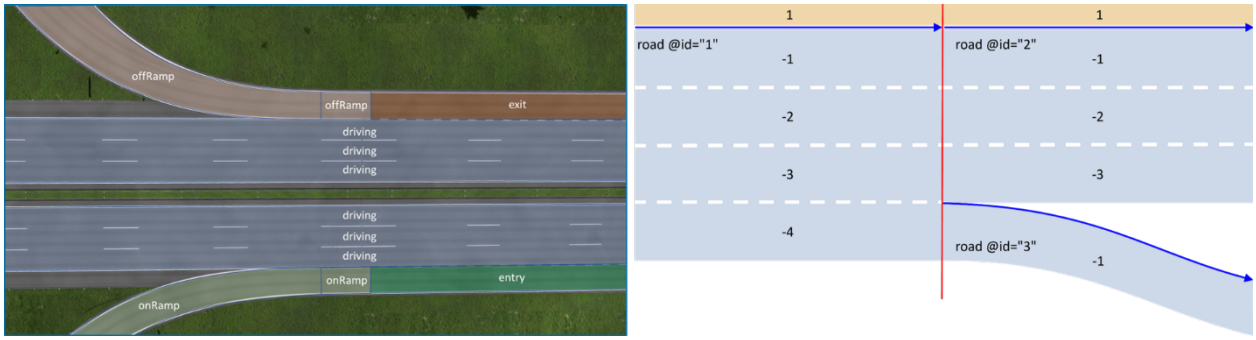
Εικόνα 3: Αριστερά, γραφική αναπαράσταση των πληροφοριών για τους δρόμους που κωδικοποιούνται στο αρχείο OpenDRIVE. Δεξιά, η χρήση των road ids σε έναν περίπλοκο κόμβο.

Εκτός από σημασιολογικά δεδομένα, οι ετικέτες των λωρίδων και άλλων δομικών στοιχείων του δικτύου φέρουν μέσω κατάλληλης ιδιότητας, μοναδικό αναγνωριστικό id. Αυτό είναι εξόχως χρήσιμο για την περιγραφή των χωροταξικών σχέσεων μεταξύ δρόμων, αλλά και όπως θα εξηγηθεί στην συνέχεια για την περιγραφή των θέσεων των οντοτήτων σε ένα σενάριο κίνησης.

Διασταυρώσεις

Η διασταύρωση (junction) στο πρότυπο OpenDRIVE περιγράφει οποιαδήποτε σύνδεση μεταξύ δύο ή περισσοτέρων δρόμων. Αναγνωρίζονται τέσσερεις βασικές κατηγορίες διασταυρώσεων:

- Common (κοινή): η συνήθης διασταύρωση όπου δύο δρόμοι τέμνονται.
- Direct (άμεση): διασταύρωση όπου τα κινούμενα οχήματα μπορούν να αλλάξουν δρόμο, όμως δεν διασταυρώνονται με το άλλο κινούμενο ρεύμα. Παράδειγμα τέτοιας είναι δύο δρόμοι που συμβάλλουν.
- Virtual (εικονική): ο κυρίως δρόμος δεν διακόπτεται. Παράδειγμα τέτοιας παράδρομος για είσοδο σε ιδιωτικό χώρο ή περιοχή στάθμευσης παράπλευρη στον δρόμο.
- Crossings (διάβαση): διασταύρωση με διαβάσεις πεζών ή σιδηροτροχιές.



Εικόνα 4: Αριστερά, μία κατηγορία λωρίδας που απαντάται σε σενάριο της εργασίας. Δεξιά, παράδειγμα μίας direct junction.

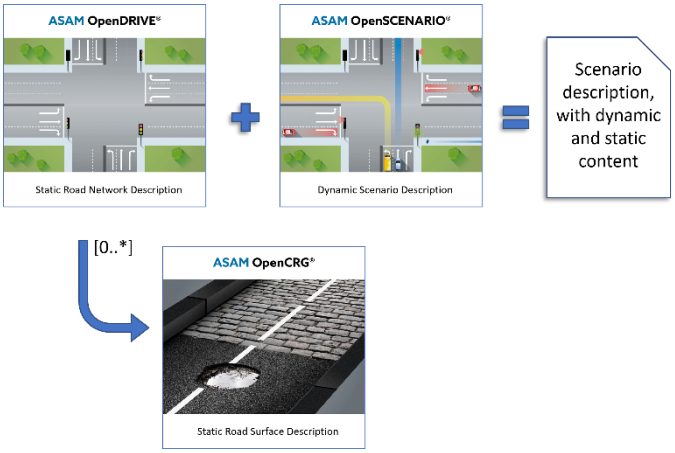
Το πρότυπο δύναται επίσης να περιγράψει σήραγγες και γέφυρες. Στο πλαίσιο της εργασίας χρησιμοποιήθηκαν σχετικά απλοί ισόπεδοι χάρτες που δεν αξιοποιούν τις σύνθετες αυτές δυνατότητες του σεναρίου.

```
<road name="Road 1" length="3.98690349181210 31e+02" id="1" junction="-1">
    <type s="0.0000000000000000e+00" type="town">
        <speed max="65" unit="mph"/>
    </type>
    <planView>
        <geometry s="0.0000000000000000e+00" x="-2.9979000854492188e+02" y="-1.0599999427795410e+00" hdg="1.304270
            <line/>
        </geometry>
    </planView>
    <elevationProfile>
        <elevation s="0.0000000000000000e+00" a="0.0000000000000000e+00" b="0.0000000000000000e+00" c="0.000000000
    </elevationProfile>
    <lateralProfile>
        <superelevation s="0.0000000000000000e+00" a="0.0000000000000000e+00" b="0.0000000000000000e+00" c="0.0000
        <shape s="0.0000000000000000e+00" t="-5.0000000000000000e-01" a="0.0000000000000000e+00" b="0.000000000000
    </lateralProfile>
    <lanes>
        <laneOffset s="0.0000000000000000e+00" a="0.0000000000000000e+00" b="0.0000000000000000e+00" c="0.00000000
        <laneOffset s="2.4117999999999998e+02" a="0.0000000000000000e+00" b="0.0000000000000000e+00" c="0.00000000
        <laneOffset s="2.7286000000000001e+02" a="0.0000000000000000e+00" b="0.0000000000000000e+00" c="0.00000000
        <laneOffset s="2.8369999999999999e+02" a="0.0000000000000000e+00" b="0.0000000000000000e+00" c="0.00000000
        <laneSection s="0.0000000000000000e+00" singleSide="false">
            <left>
                <lane id="3" type="driving" level="false">
```

Εικόνα 5: Χαρακτηριστικό απόσπασμα αρχείου .xodr: για τον δρόμο ορίζονται αρχικώς βασικές πληροφορίες και στην συνέχεια περιγράφεται γεωμετρικά και ανά λωρίδα.

2.2.2 OpenSCENARIO

Η προτυποποίηση του τρόπου περιγραφής σεναρίων είναι απαραίτητη, ούτως ώστε να μπορεί μεγάλος αριθμός σημαντικών καταστάσεων προσομοίωσης να εξετασθεί σε διάφορους προσομοιωτές με μηδαμινή ανάγκη προσαρμογής στον κάθε έναν. Το πρότυπο ASAM OpenSCENARIO καθορίζει τις προδιαγραφές και την μορφή αρχείου για την περιγραφή του δυναμικού περιεχομένου προσομοιωτών οδήγησης. Η βασική περίπτωση χρήσης για την οποία προορίζεται είναι η καταγραφή περίπλοκων, σύγχρονων κινήσεων που εμπλέκουν πολλές οντότητες, όπως οχήματα, πεζούς και άλλα κινούμενα σώματα [12].

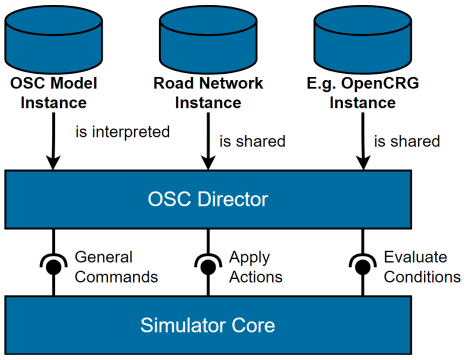


Εικόνα 6: Διαγραμματική απεικόνιση της συνέργειας των προτύπων του ASAM στην σύνθεση ολοκληρωμένων σεναρίων.

Σχεδιασμός

Το πρότυπο αναλύει την περιγραφή του σεναρίου σε δύο στάδια: τον σχεδιασμό (design time) και την εκτέλεση (runtime). Ο σχεδιασμός γίνεται βάσει της γλώσσας και του συντακτικού που ορίζονται για τα αρχεία OpenSCENARIO, ενώ η περιγραφή της εκτέλεσης εξασφαλίζεται μέσω κάποιων κανόνων με τους οποίους οφείλει να συμμορφώνεται ο προσομοιωτής που θα αναλάβει να εκτελέσει το σενάριο [12].

Το διάγραμμα απεικονίζει τα απαραίτητα συστατικά μέρη του συστήματος προσομοίωσης βάσει OpenSCENARIO. Η πλήρης περιγραφή του σεναρίου προκύπτει ως σύνθεση των OpenSCENARIO Model Instance, Road Network Instance και OpenCRG Instance. Για την ερμηνεία και πραγμάτωση αυτής από τον πυρήνα του προσομοιωτή (Simulator Core), μεσολαβεί ο ενορχηστρωτής OSC Director, ένα ενδιάμεσο λογισμικό (middleware) υπεύθυνο για την μεταγλώττιση του σεναρίου σε εντολές εκτελέσιμες από την μηχανή φυσικής του προσομοιωτή.



Εικόνα 7: Η αρχιτεκτονική λογισμικού ενός προσομοιωτή βασισμένου στα τα πρότυπα ASAM.

Στην πραγματικότητα αναπτύσσονται παράλληλα δύο πρότυπα για τον ίδιο σκοπό, το βασισμένο στην XML OpenSCENARIO 1.0, ονομαζόμενο και “OpenSCENARIO XML”, και

το 2.0 ή “OpenSCENARIO DSL”, το οποίο ορίζει μία ειδική γλώσσα (domain-specific language) ώστε να επιτύχει υψηλότερο επίπεδο αφαίρεσης. Το πρακτικό μέρος της παρούσης εργασίας πραγματοποιήθηκε με το πρώτο εκ των δύο, του οποίου η τελευταία έκδοση είναι η 1.3.0. Στην συνέχεια παρουσιάζεται το πρότυπο 1, ενώ τα πρακτικά ζητήματα συμβατότητάς του με τον προσομοιωτή εξηγούνται στο Κεφάλαιο 4. Η παρουσίαση είναι αρκετά αναλυτική και εκτενής διότι, παρά το γεγονός πως τα εργαλεία που χρησιμοποιούνται στην εργασία χειρίζονται τα αρχεία .xosc αυτομάτως, στην πραγματικότητα χρειάστηκε μεγάλη εξοικείωση με την σύνταξη των αρχείων για να αντιμετωπισθούν διάφορα προβλήματα στην πορεία.

OpenSCENARIO XML

Η κωδικοποίηση των σεναρίων γίνεται με τον ορισμό ιεραρχικής δομής στοιχείων, κάθε ένα εκ των οποίων αναπαριστά με σαφή τρόπο (όχι δηλαδή αφηρημένο) ένα δομικό μέρος του σεναρίου. Περιληπτικά, κάθε αρχείο OpenSCENARIO 1.0 περιέχει:

- Καθορισμό του λογικού οδικού δικτύου όπου θα λάβει χώρα το σενάριο.
- Αρχικοποίηση των συμμετεχουσών στην δράση οντοτήτων, ήτοι κυρίως οχημάτων και πεζών.
- Ακριβή περιγραφή των γεγονότων που συμβαίνουν στο σενάριο, και των συνθηκών έναρξης και λήξης κάθε ενός.

Η διάρθρωση ενός σεναρίου αποτυπωμένου στο πρότυπο μπορεί επομένως να συνοψιστεί ως εξής: Οντότητες (`<Entity>`) πραγματοποιούν δράσεις (`<Action>`) ελεγχόμενες από συνθήκες (`<Trigger>`), εντός ενός οδικού δικτύου (`<RoadNetwork>`). Στην συνέχεια παρουσιάζονται αναλυτικά οι προδιαγραφές του προτύπου για κάθε ένα από τα συστατικά αυτά μέρη.

Οδικό δίκτυο

Η ετικέτα `<RoadNetwork>` περιέχει αναφορά στο λογικό οδικό δίκτυο (`<LogicFile>`) όπου θα λάβει χώρα το σενάριο. Η αναφορά αυτή είναι στην απλούστερη περίπτωση προς την αντίστοιχη περιγραφή του δικτύου σε OpenDRIVE, δηλαδή ένα αρχείο τύπου .xodr. Αν χρειάζεται να συμπεριληφθούν ειδικά τρισδιάστατα μοντέλα στο προσομοιούμενο περιβάλλον, υπάρχει η δυνατότητα να γίνει σύνδεση και με βιβλιοθήκες για τρισδιάστατα γραφικά όπως η OpenSceneGraph [24]. Βέβαια, κάθε προσομοιωτής έχει την ελευθερία να ερμηνεύει το στοιχείο αυτό με τον κατάλληλο τρόπο για την εκάστοτε υλοποίηση του χάρτη στην μηχανή γραφικών.

Συχνά είναι αναγκαία ή ιδιαιτέρως εξυπηρετική η αναφορά εντός του σεναρίου σε συγκεκριμένα στοιχεία του οδικού δικτύου, όπως μία συγκεκριμένη λωρίδα ενός δρόμου, ώστε να κατευθυνθεί σε αυτήν κάποιο όχημα. Αυτήν την δυνατότητα την παρέχει η διαλειτουργικότητα μεταξύ OpenDRIVE και OpenSCENARIO. Μέσα από το OpenSCENARIO μπορούν να γίνονται αναφορές σε αναγνωριστικά περιεχομένων του προκαθορισμένου χάρτη OpenDRIVE [12].

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSCENARIO>
    <FileHeader revMajor="1" revMinor="0" date="2023-12-12T13:01:01" description="Exported from RoadRunner versionR202
    <ParameterDeclarations/>
    <CatalogLocations/>
    <RoadNetwork>
        <LogicFile filepath="./Motorway.xodr"/>
        <SceneGraphFile filepath="./Motorway.osgb"/>
    </RoadNetwork>
```

Εικόνα 8: Οι πρώτες σειρές ενός αρχείου .xosc όπου μεταξύ άλλων ορίζεται ο χάρτης.

Οντότητες

Οι συμμετέχουσες στην δράση οντότητες αρχικοποιούνται μέσα στην ετικέτα `<Entities>`. Εκτός από οχήματα και πεζούς, το πρότυπο προβλέπει την δυνατότητα εισαγωγής ως οντότητες διαφόρων αντικειμένων (όπως δένδρα ή στύλους) που δεν περιέχονται εξ αρχής στον χάρτη. Τα διαφορετικά είδη οντοτήτων φέρουν και διαφορετικές ιδιότητες. Για κάθε όχημα `<Vehicle>` καθορίζεται μοναδικό αναγνωριστικό, κατηγορία οχήματος (όπως επιβατηγό, δίκυκλο, φορτηγό), όνομα μοντέλου (και αυτό εξαρτάται από την υλοποίηση και ορολογία του εκάστοτε προσομοιωτή), και επιπλέον στοιχεία για τις διαστάσεις του chassis (`<BoundingBox>`), των αξόνων και των τροχών (`<Axles>`), και τις επιδόσεις που επιτυγχάνει (`<Performance>`) [12].

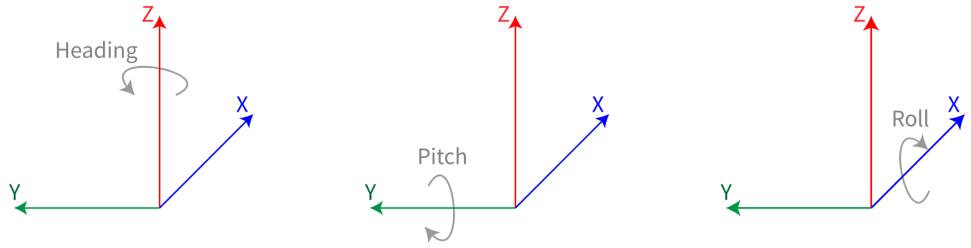
```
<ScenarioObject name="Sedan">
    <Vehicle name="Sedan" vehicleCategory="car">
        <ParameterDeclarations/>
        <BoundingBox>
            <Center x="-0.00706172" y="5.96046e-08" z="0.822382"/>
            <Dimensions height="1.264" length="4.55561" width="1.93778"/>
        </BoundingBox>
        <Performance maxAcceleration="5" maxDeceleration="5" maxSpeed="65"/>
        <Axles>
            <FrontAxle maxSteering="0.698132" positionX="1.38385" positionZ="0.34415" trackWidth="1.13434" whe
            <RearAxle maxSteering="0" positionX="-1.32946" positionZ="0.34415" trackWidth="1.13434" wheelDiame
        </Axles>
        <Properties/>
    </Vehicle>
</ScenarioObject>
```

Εικόνα 9: Ετικέτα Vehicle και τα περιεχόμενα αυτής σε αρχείο .xosc.

Storyboard

Το σενάριο περιέχει ένα μοναδικό `<Storyboard>`, δηλαδή γενικό σχεδιάγραμμα που περιγράφει πλήρως την σειρά των γεγονότων, καθορίζοντας για κάθε ένα πώς και πότε συμβαίνει, και ποιοί παράγοντες συμμετέχουν σε αυτό.

Το Storyboard αποτελείται από δύο μέρη: `<Init>` και `<Story>`. Στο πρώτο προσδιορίζονται οι αρχικές καταστάσεις των στοιχείων του σεναρίου, όπως αρχικές θέσεις, προσανατολισμοί και ταχύτητες των Entities που έχουν δηλωθεί πρότερα. Κάθε τέτοια αρχικοποίηση αποτυπώνεται με μία ετικέτα `<Action>`. Αυτό το τμήμα του Storyboard έχει αμιγώς δηλωτικό χαρακτήρα, δεν μπορούν δηλαδή να τοποθετηθούν σε αυτό λογικές συνθήκες για την τήρηση ή μη των κανόνων, ούτε για την συμπεριφορά των οντοτήτων.



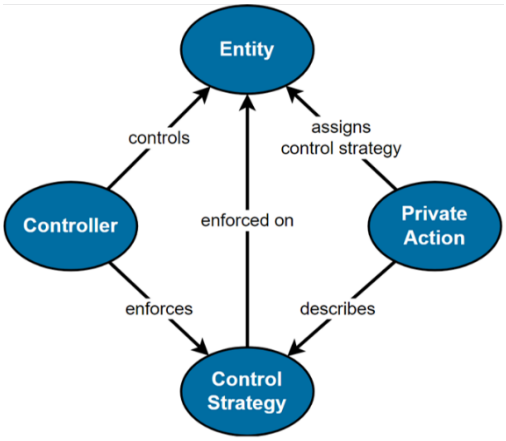
Εικόνα 10: Το σύστημα συντεταγμένων και προσανατολισμού κατά ISO 8855 [26].

Η έννοια της δράσης (“Action”) στο πρότυπο είναι αρκετά ευρεία και αναφέρεται γενικώς στην υπαγόρευση της συμπεριφοράς ή της καταστάσεως των στοιχείων του σεναρίου. Χαρακτηριστικό παραδείγματα Action στην πράξη είναι η αλλαγή της ταχύτητας ενός οχήματος. Οι Actions διακρίνονται σε δύο βασικές υποκατηγορίες, Global και Private. Η πρώτη χρησιμοποιείται για τροποποίηση καθολικών συνθηκών του σεναρίου που δεν αναφέρονται σε συγκεκριμένη οντότητα, όπως οι καιρικές συνθήκες του περιβάλλοντος και η ώρα της ημέρας, ενώ η δεύτερη εφαρμόζεται και επηρεάζει μεμονωμένες Entities. Οι ετικέτες Action εμφανίζονται και στις δύο περιοχές του Storyboard: στην μεν Init αρχικοποιούν την διάταξη του σεναρίου, στην δε Story όπως εξηγείται στην συνέχεια τροποποιούν βάσει συνθηκών τις παραμέτρους κατάλληλα ώστε να εκτυλιχθεί η ροή του σεναρίου.

```
<Storyboard>
    <Init>
        <Actions>
            <GlobalAction>
                <EnvironmentAction>
                    <Environment name="Environment1">
                        <TimeOfDay animation="false" dateTime="2021-03-25T12:00:00"/>
                        <Weather cloudState="free">
                            <Sun intensity="0.85" azimuth="0" elevation="1.31"/>
                            <Fog visualRange="100000.0"/>
                            <Precipitation precipitationType="dry" intensity="0.0"/>
                        </Weather>
                       <RoadCondition frictionScaleFactor="1.0"/>
                    </Environment>
                </EnvironmentAction>
            </GlobalAction>
            <Private entityRef="Sedan">
                <PrivateAction>
                    <TeleportAction>
                        <Position>
                            <LanePosition roadId="1" laneId="2" offset="5.41974e-14" s="396.501">
                                <Orientation type="absolute" h="-3.14029" p="0" r="0"/>
                            </LanePosition>
                        </Position>
                    </TeleportAction>
                </PrivateAction>
```

Εικόνα 11: Global και Private Actions.

Μία ειδική και ιδιαιτέρως χρήσιμη PrivateAction που προβλέπει το πρότυπο είναι η `<AssignControllerAction>`. Μέσω αυτής ανατίθεται σε μία συγκεκριμένη Entity ένα οδηγικό μοντέλο, το οποίο στο πλαίσιο της εργασίας λειτουργεί ως διεπαφή με την φυσική είσοδο ώστε το όχημα αυτό να ελέγχεται από τον χρήστη. Γενικώς η «στρατηγική ελέγχου» (control strategy) που προσδίδεται σε οιανδήποτε οντότητα δεν περιγράφεται αυστηρώς από το πρότυπο αλλά η υλοποίησή της επαφίεται στον εκάστοτε προσομοιωτή που εκτελεί το σενάριο.



Εικόνα 12: Η αλληλεπίδραση μεταξύ οντοτήτων, δράσεων και ελεγκτών.

Story

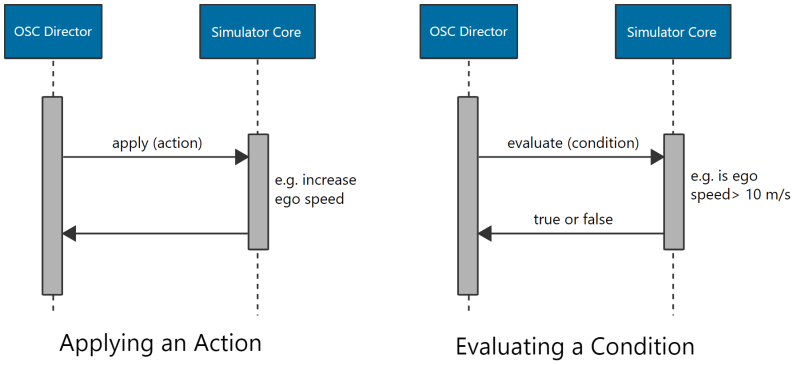
Στο δεύτερο μέρος του Storyboard, το <Story>, καταστρώνεται με ιεραρχική δομή η ακολουθία των γεγονότων που απαρτίζουν το σενάριο. Σε αυτό το σημείο, προτού παρουσιασθούν τα περιεχόμενα του <Story>, πρέπει να εξηγηθεί ο μηχανισμός που ορίζει το πρότυπο για τον έλεγχο της έναρξης και της λήξης κάθε δράσης στο σενάριο, οι Triggers. Κάθε ετικέτα <Trigger> συνδέεται με κάποιο στοιχείο της <Story> και περιγράφει τις συνθήκες ενεργοποίησης αυτού. Συγκεκριμένα, περιέχει μία ή περισσότερες ομάδες συνθηκών <ConditionGroup>, κάθε μια εκ τω οποίων περιέχει με την σειρά της μία ή περισσότερες ετικέτες <Condition>, που αναπαριστούν μία λογική συνθήκη. Οι Conditions διακρίνονται αφ' ενός σε <ByEntityCondition>, που αφορούν την κατάσταση μίας οντότητας ή την σχέση μεταξύ οντοτήτων (όπως ταχύτητα οχήματος ή απόσταση μεταξύ οχημάτων), αφ' ετέρου σε <ByValueCondition>, που αφορούν δεδομένα της προσομοίωσης, όπως την χρονική διάρκεια εκτέλεσης. Μία <ConditionGroup> αποτιμάται αληθής όταν η λογική σύζευξη όλων των <Condition> που περιέχει αποτιμάται αληθής, και τελικώς η <Trigger> πυροδοτείται όταν η λογική διάζευξη όλων των <ConditionGroup> που περιέχει αποτιμάται αληθής. Τότε, η δράση του σεναρίου με την οποία συσχετίζεται εκκινείται, αν πρόκειται για <StartTrigger>, ή τερματίζεται αν πρόκειται για <StopTrigger>.

```
<StopTrigger>
    <ConditionGroup>
        <Condition name="Distance_To_Actor" conditionEdge="none" delay="0.0000000000000000e+00">
            <ByEntityCondition>
                <TriggeringEntities triggeringEntitiesRule="any">
                    <EntityRef entityRef="Sedan"/>
                </TriggeringEntities>
                <EntityCondition>
                    <RelativeDistanceCondition entityRef="CompactCar" relativeDistanceType="cartesianDistance" value="10" freespace="true" rule="greaterThan"/>
                </EntityCondition>
            </ByEntityCondition>
        </Condition>
    </ConditionGroup>
</StopTrigger>
```

Εικόνα 13: Παράδειγμα συνθήκης εν σχέσει με άλλη οντότητα. Τα φυσικά μεγέθη μετρούνται στο σύστημα S.I.

Στην πράξη οι Triggers λειτουργούν μεταβάλλοντας την κατάσταση του «κύκλου ζωής» - “life cycle state” του στοιχείου (<StoryboardElement>) με το οποίο συνδέονται. Το πρότυπο ορίζει τις εξής καταστάσεις:

- `initState` και `endState`: εξαρτώνται από την υλοποίηση του προσομοιωτή (implementation-specific) και αναπαριστούν την δημιουργία και την καταστροφή αντιστοίχως του StoryboardElement.
- `standbyState`: Το στοιχείο βρίσκεται σε αναμονή να ενεργοποιηθεί η StartTrigger του.
- `runningState`: Οι δράσεις που υπαγορεύονται από το στοιχείο είναι υπό εκτέλεση.
- `completeState`: Οι δράσεις που υπαγορεύονται από το στοιχείο έχουν ολοκληρωθεί ή διακοπεί.



Εικόνα 14: Διάγραμμα interaction (αλληλεπίδρασης) μεταξύ προσομοιωτή και scenario director.

Περιγραφή της δράσης

Όπως έχει αναφερθεί, η ατομική μονάδα που περιγράφει μία δράση κάποιας οντότητας στο σενάριο είναι η `<Action>`. Μερικές από τις συχνότερα χρησιμοποιούμενες Actions είναι:

- `TeleportAction`: μεταφορά οχήματος σε συντεταγμένες
- `SpeedAction`: αλλαγή ταχύτητας
- `LaneChangeAction`: αλλαγή λωρίδας
- `FollowTrajectoryAction`: κίνηση βάσει καθορισμένης τροχιάς

Εντός της `<Story>`, οι Actions ομαδοποιούνται εντός των ετικετών `<Event>`. Η ετικέτα `<Event>` πρόκειται απλώς για περιέκτη (container) πολλών Actions, μαζί με μία StartTrigger. Πολλά Events ομαδοποιούνται περαιτέρω σε μία Maneuver. Από το πρότυπο παρέχεται στην ετικέτα `<Maneuver>` η δυνατότητα καθορισμού της προτεραιότητας που θα έχουν τα Events που αυτή περιέχει. Κάθε ετικέτα `<Event>` διαθέτει το πεδίο `priority`, το οποίο λαμβάνει τις εξής τιμές:

- `override`: όλα τα υπόλοιπα Events εντός της κοινής Maneuver παύονται όταν ενεργοποιηθεί το Event ώστε να εκτελεσθεί αυτό.
- `parallel`: το Event εκτελείται παράλληλα (ανεξάρτητα) από τα άλλα Events της Maneuver.

Επιπλέον ορίζεται περιέκτης και για Manuevers, η ετικέτα `<ManeuverGroup>`. Μία ManeuverGroup περιέχει μία ή περισσότερες Maneuvers, και επίσης καθορίζει ποιές οντότητες (Entities) του σεναρίου συμμετέχουν σε αυτές, μέσω της ετικέτας `<Actors>`. Ως προς την ιδιότητα αυτήν, είναι μοναδική μεταξύ των ετικετών που εσωκλείουν την δράση. Τέλος, οι ManeuverGroups εντάσσονται εντός ετικετών `<Act>`, για τις οποίες ορίζονται Start και

StopTriggers. Ο χωρισμός των γεγονότων σε Acts υποδηλώνει την χρονική σειρά αυτών στο σενάριο. Αυτή η εκτεταμένη χρήση δομών - περιεκτών προσδίδει ευελιξία στον καθορισμό Triggers για σύνθετα σύνολα γεγονότων, αλλά και καθιερώνει ουσιώδη σημασιολογική διάκριση μεταξύ των διαφόρων επιπέδων: ένα Event περιγράφει πιό σύνθετη κίνηση από τις επιμέρους Actions, η Maneuver εισαγάγει την παράμετρο της προτεραιότητας μεταξύ Events, και ούτω καθ' εξής.

```
<Act name="Act2">
    <ManeuverGroup name="Act2_SedanGroup" maximumExecutionCount="1">
        <Actors selectTriggeringEntities="false">
            <EntityRef entityRef="Sedan"/>
        </Actors>
        <Maneuver name="Act2_Maneuver">
            <Event name="Act2_Event" priority="parallel">
                <Action name="Act2_SedanGroup_Change_Lane">
                    <PrivateAction>
                        <LateralAction>
                            <LaneChangeAction>
                                <LaneChangeActionDynamics dynamicsShape="linear" dynamicsDimension="distance" value="80"/>
                                <LaneChangeTarget>
                                    <RelativeTargetLane entityRef="Sedan" value="-1"/>
                                </LaneChangeTarget>
                            </LaneChangeAction>
                        </LateralAction>
                    </PrivateAction>
                </Action>
                <StartTrigger>
                    <ConditionGroup>
                        <Condition name="Phase_State" conditionEdge="none" delay="0.0000000000000000e+00">
                            <ByValueCondition>
                                <StoryboardElementStateCondition storyboardElementType="act" storyboardElementRef="Act" state="completeState"/>
                            </ByValueCondition>
                        </Condition>
                    </ConditionGroup>
                </StartTrigger>
            </Event>
```

Εικόνα 15: Απόσπασμα από ένα Act όπου φαίνεται η ιεραρχία των δομών αναπαράστασης των δράσεων.

Συνοψίζοντας, η δράση των κινουμένων σωμάτων στο σενάριο περιγράφεται στο μέρος του αρχείου OpenSCENARIO που περικλείεται από την ετικέτα `<Story>`. Αυτή περιέχει Acts, τα οποία ορίζουν το «πότε» για κάθε γεγονός στο σενάριο. Κάθε Act περιέχει ManeuverGroups, τα οποία με την σειρά τους καθορίζουν το «ποιός», και οι Maneuvers που τις απαρτίζουν απαντούν στο «τί». Η περιγραφή επακριβώς των κινήσεων σε χαμηλό επίπεδο με όρους φυσικής γίνεται από τις Actions εντός των Events, όταν αυτές ενεργοποιούνται βάσει προκαθορισμένων από τον χρήστη συνθηκών [12].

Χάραξη διαδρομής

Η προεπιλεγμένη από το πρότυπο διαδρομή που ακολουθεί μία Entity στην οποία δίνεται ταχύτητα είναι «ευθεία σε κάθε διασταύρωση». Με τον λέξη «ευθεία» εννοείται ότι, εφ' όσον το σενάριο δεν επιβάλλει αλλαγή, το αυτοκίνητο συνεχίζει σταθερά στον ίδιο δρόμο και στην ίδια λωρίδα, και αν ο δρόμος φθάσει σε διασταύρωση, αν μπορεί συνεχίζει ευθεία, ενώ αν δεν μπορεί τότε στρίβει προς την κατεύθυνση που επιτρέπεται. Αν επιτρέπεται να στρίψει και προς τις δύο κατευθύνσεις, τότε η προεπιλεγμένη στροφή είναι προς την πλευρά οδήγησης του δρόμου. Αυτή η πολιτική προβλέπεται ούτως ώστε να εξασφαλίζεται αναπαράξιμη συμπεριφορά σε διαφορετικούς προσομοιωτές.

Για τις περιπτώσεις όπου η αυτόματη αυτή συμπεριφορά δεν καλύπτει τις ανάγκες του σεναρίου, προβλέπονται μέθοδοι ακριβέστερης χάραξης διαδρομής, είτε καθορίζοντας ένα διαφορετικό μοντέλο συμπεριφοράς, είτε καθορίζοντας επακριβώς την επιθυμητή τροχιά (Trajectory) του αυτοκινήτου. Στην δεύτερη περίπτωση, η τροχιά αναπαρίσταται από

μαθηματική καμπύλη που διέρχεται από σύνολο σημείων αναφοράς (Waypoints), και συνοδεύεται από δεδομένα για την επιθυμητή ταχύτητα ή χρόνο άφιξης σε κάθε σημείο. Η ετικέτα `<Trajectory>` αποτελείται από σειρά ετικετών `<Vertex>` (κορυφών). Κάθε κορυφή αναπαριστά μία στιγμή στον χρόνο και μία θέση στον χώρο. Η Vertex διαθέτει πεδίο time, το οποίο καθορίζει την χρονική στιγμή που θε βρεθεί στην θέση εκείνη το σώμα, και περιέχει ετικέτα `<Position>`, στην οποία καθορίζονται οι συντεταγμένες στον χώρο.

```
<RoutingAction>
    <FollowTrajectoryAction>
        <Trajectory name="Sedan_Trajectory" closed="false">
            <Shape>
                <Polyline>
                    <Vertex time="0.0000000000000000e+00">
                        <Position>
                            <WorldPosition x="-4.6474508178254126e+01" y="-3.3610822034303816e+00" z="0.0000000000000
                        </Position>
                    </Vertex>
                    <Vertex time="1.1857096035600095e-01">
                        <Position>
                            <WorldPosition x="-4.5894492235670420e+01" y="-3.4885953128257894e+00" z="0.0000000000000
                        </Position>
                    </Vertex>
                    <Vertex time="3.2521098663616699e-01">
                        <Position>
                            <WorldPosition x="-4.4815575473890895e+01" y="-3.7393921539743431e+00" z="0.0000000000000
                        </Position>
                    </Vertex>
```

Εικόνα 16: Απόσπασμα από ετικέτα FollowTrajectoryAction και το περιεχόμενό της.

Το πρότυπο περιέχει αρκετές ακόμη σύνθετες έννοιες και δομές των οποίων η παρουσίαση υπερβαίνει την θεωρητική θεμελίωση αυτής της εργασίας. Έχει ήδη εξηγηθεί ότι για ορισμένα θέματα της υλοποίησης και εκτέλεσης του σεναρίου, το πρότυπο αφήνει την επιλογή στον προσομοιωτή. Επιπλέον αυτών που έχουν αναφερθεί, δεν περιέχει έννοιες για τον καθορισμό ελέγχων (test configuration description) ή την αξιολόγηση των αποτελεσμάτων αυτών. Τέλος, δεν περιέχει μοντέλα συμπεριφοράς του οδηγού (driver models), αλλά όπως αναφέρθηκε δίνει την δυνατότητα σύνδεσης μίας οντότητας με κάποιο μοντέλο που υλοποιείται από τον εκάστοτε προσομοιωτή.

Κεφάλαιο 3: Εργαλεία Ανάπτυξης

3.1 Python

Η Python είναι μία υψηλού επιπέδου διερμηνευόμενη (interpreted) γλώσσα προγραμματισμού ανοικτού κώδικα που συνδυάζει ("multi-paradigm") στοιχεία διαδικαστικού (procedural), αντικειμενοστρεφούς (object-oriented) και συναρτησιακού (functional) προγραμματισμού. Χρησιμοποιείται ευρύτατα σε διάφορα είδη εφαρμογών χάρη στην ευκολία, αναγνωσιμότητα και παραγωγικότητα που παρέχει. Το αντίτιμο για αυτά τα προτερήματα είναι οι χαμηλές επιδόσεις σε ταχύτητα. Ένας ακόμη βασικός λόγος για την εξαιρετικά ευρεία διάδοσή της είναι η μεγάλη ποικιλία τρίτων βιβλιοθηκών (modules) που διατίθενται για αυτήν [27]. Στην εργασία η Python χρησιμοποιείται σε τρία σημεία:

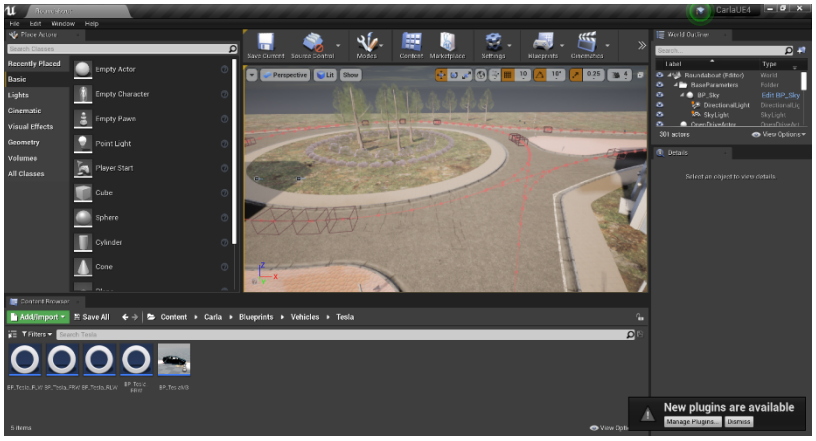
- Βασικά εξαρτήματα του προσομοιωτή CARLA είναι υλοποιημένα σε Python (συγκεκριμένα την έκδοση 3.8 της γλώσσας) με την εξωτερική βιβλιοθήκη PyGame.
- Έχει επιλεγεί για την συγγραφή των scripts αυτοματοποίησης παραγωγής σεναρίων. Αυτό επειδή χάρη στις εξωτερικές βιβλιοθήκες grpcio και grpcio-tools καθίσταται δυνατή και σχετικά απλή η επικοινωνία της Python μέσω gRPC με το αντίστοιχο API του RoadRunner. Επιπλέον για τα scripts αυτά φάνηκε ιδιαιτέρως χρήσιμη η βιβλιοθήκη jupyter, με την οποία δημιουργούνται jupyter notebooks, διαδραστικά αρχεία όπου ο κώδικας δομείται σπονδυλωτά σε κελλιά (cells) εκτελούμενα χωριστά το ένα από το άλλο αλλά με κοινή μνήμη [45].
- Με την ενσωματωμένη βιβλιοθήκη xml.etree.ElementTree [28] επιτρέπει την ανάλυση και επεξεργασία αρχείων XML, και ως εκ τούτου έχει χρησιμοποιηθεί για την συγγραφή βοηθητικών scripts επεξεργασίας των σεναρίων OpenSCENARIO XML.

3.2 Unreal Engine 4

Η Unreal Engine είναι λογισμικό (game engine) για τρισδιάστατα γραφικά που αναπτύσσεται από την Epic Games και χρησιμοποιείται κυρίως για video games, αλλά βρίσκει εφαρμογή και σε προσομοιωτές. Πρόκειται για μία από τις ευρύτερα διαδεδομένες μηχανές γραφικών, καθώς μπορεί να εγκατασταθεί σε όλα σχεδόν τα σύγχρονα λειτουργικά συστήματα. Ο προσομοιωτής CARLA βασίζεται στην τέταρτη έκδοσή της, την Unreal Engine 4, η οποία είναι ανοικτού κώδικα και διατίθεται δωρεάν για μη εμπορική χρήση [29].

Επειδή ο πηγαίος κώδικας της Unreal είναι γραμμένος σε C++, για την εγκατάστασή της σε λειτουργικό σύστημα Windows απαιτείται το πακέτο ανάπτυξης εφαρμογών "Microsoft Visual Studio Build Tools". Αυτό περιέχει μεταγλωττιστή C++ και άλλα εργαλεία που επιτρέπουν την μεταγλώττιση του κώδικα της Unreal και την εκτέλεση του προγράμματος Unreal Editor. Με τον Unreal Editor μπορούν να γίνουν τροποποιήσεις και προσθήκες στην βιβλιοθήκη γραφικών του CARLA (βλ. Παράρτημα Β').

Μία αξιοσημείωτη διαφορά της Unreal μεταξύ Windows και Linux (τα δύο συστήματα στα οποία αναπτύσσεται και τρέχει ο CARLA) είναι η διεπαφή με την μονάδα γραφικών του υπολογιστή. Σε Windows χρησιμοποιείται η DirectX, ενώ σε Linux η OpenGL. Ωστόσο, η διαφορά αυτή δεν δημιουργεί προβλήματα συμβατότητας μεταξύ των συστημάτων ως προς την λειτουργία του CARLA και την εκτέλεση σεναρίων.



Εικόνα 17: Επιθεώρηση εισηγμένων γραφικών στον Unreal editor.

3.3 CARLA

Ο προσομοιωτής ανοικτού κώδικα CARLA (“Car Learning to Act”) αναπτύχθηκε με βάση την Unreal Engine 4, και επικεντρώνεται στην προσομοίωση οδήγησης σε αστικό περιβάλλον με αληθοφανή γραφικά. Ο κύριος σκοπός του έργου είναι ο σχεδιασμός, η εκπαίδευση, μέτρηση απόδοσης και η επικύρωση μοντέλων αυτόνομης οδήγησης, τόσο στο επίπεδο της αντίληψης (perception) όσο και στου ελέγχου (control). Ως εκ τούτου, δεν επικεντρώνεται στην πιστή αναπαραγωγή της λειτουργίας του αυτοκινήτου από μηχανολογική σκοπιά. Περιέχει έτοιμη βιβλιοθήκη τρισδιάστατων γραφικών αντικειμένων (οχημάτων, κτηρίων, πινακίδων και λοιπών στοιχείων του αστικού οδικού δικτύου) ειδικά κατασκευασμένων για τις ανάγκες της προσομοίωσης. Για την ρύθμιση της συμπεριφοράς των οντοτήτων που δεν είναι υπό τον έλεγχο του χρήστη (των non-player characters - NPCs), ο CARLA χρησιμοποιεί το τυποποιημένο μοντέλο συμπεριφοράς οχημάτων της Unreal, το PhysXVehicles, με κάποιες τροποποιήσεις ώστε τα οχήματα να υπακούουν στους οδικούς κανόνες και να αποφεύγουν συγκρούσεις. Επίσης, παρέχει εικονικά σήματα από προσομοιωμένους αισθητήρες του αυτοκινήτου, όπως RGB cameras, RADAR και LIDAR. Επιπλέον αυτών των σημάτων, ο προσομοιωτής μπορεί ανά πάσα στιγμή να δίνει πληροφορίες για τα φυσικά μεγέθη και καταστάσεις του εικονικού κόσμου, όπως αυτές προκύπτουν από την αλληλεπίδραση με την μηχανή φυσικής της Unreal. Τέτοιες πληροφορίες είναι η ταχύτητα, η επιτάχυνση, η ακριβής θέση κάθε οχήματος και τα επίπεδα ζημιάς από συγκρούσεις. Οι πληροφορίες αυτές συνδυάζονται με σημασιολογικά δεδομένα του οδικού δικτύου ώστε ο προσομοιωτής να γνωρίζει πότε ένα όχημα οδηγεί ταχύτερα από το όριο ή εκτός των διαγραμμίσεων του δρόμου, και άλλα παρόμοια χρήσιμα συμπεράσματα. Ως προς τον εικονικό κόσμο μπορεί να προσομοιώσει ποικιλία επιπέδων φωτισμού λόγω διαφοράς ώρας της ημέρας ή ατμοσφαιρικής πυκνότητας, και διάφορες καιρικές συνθήκες και τα συνεπακόλουθα αποτελέσματα αυτών στο οδόστρωμα. Τέλος, ως προς τον έλεγχο οχήματος από τον χρήστη, ο προσομοιωτής έχει ολοκληρωμένο σύστημα εισόδου που δέχεται εντολές είτε από πληκτρολόγιο είτε από ειδικό αναλογικό μέσο για προσομοιωτή οδήγησης (τιμόνι και πεντάλ για παιχνίδια υπολογιστή), και δίνει την δυνατότητα μέσω αυτών στον χρήστη να χειρίζεται το αυτοκίνητο με όλους σχεδόν τους τρόπους που θα το έκανε στην πραγματικότητα [30].

3.3.1 Scenario runner

Το scenario runner είναι το βασικό μέρος του προσομοιωτή όπου στηρίζεται η εργασία. Πρόκειται για ένα διακριτό τμήμα λογισμικού από τον κυρίως προσομοιωτή CARLA, αλλά λειτουργεί επικουρικά ως υποσύστημα αυτού στις περιπτώσεις χρήσης που περιλαμβάνουν σενάριο από εξωτερική πηγή. Ο ρόλος του είναι η ανάλυση (parsing) του αρχείου OpenSCENARIO και η εξαγωγή από αυτό των αξιοποιήσιμων δεδομένων, η μετατροπή αυτών σε δομές αναγνωρίσιμες από τον CARLA και η επικοινωνία με το Python API για την διενέργεια του σεναρίου [31].

Υλοποίηση

Ο CARLA είναι σχεδιασμένος με αρχιτεκτονική client-server. Ο server τρέχει την προσομοίωση αλληλοεπιδρώντας με την Unreal και την αποδίδει (render) γραφικά. Η επικοινωνία μεταξύ server και client γίνεται μέσω sockets από ένα API γραμμένο σε Python. Μέσω του API αυτού οι clients αποστέλλουν δύο ειδών εντολές στον server, και λαμβάνουν τις μετρήσεις των αισθητήρων. Το πρώτο είδος εντολών είναι για τον χειρισμό του οχήματος (κατεύθυνση τιμονιού, επιτάχυνση με γκάζι, πέδηση, χειρόφρενο, αλλαγή σχέσης κιβωτίου ταχυτήτων σε όπισθεν), και το δεύτερο είδος είναι «μετά-εντολές» (meta-commands) που επηρεάζουν το περιβάλλον και την συμπεριφορά του server. Τέτοιες εντολές είναι η αλλαγή των καιρικών συνθηκών και η επανεκκίνηση της προσομοίωσης [31].

Ακολουθεί αναλυτική παρουσίαση των κυριοτέρων τμημάτων κώδικα και παραμέτρων του CARLA που αφορούν την εργασία.

3.3.2 Server

Ο server του προσομοιωτή εκκινείται από το εκτελέσιμο αρχείο CarlaUE4.exe, το οποίο δημιουργείται όταν γίνεται η διαδικασία build του CARLA (βλ. Παράρτημα Β'). Βασικές παράμετροι για την εκκίνηση του server είναι οι:

- `carla-port`: Οι θύρες του τοπικού δικτύου (TCP ports) μέσω των οποίων θα επικοινωνούν server και client. Ορίζεται η θύρα όπου ο server θα αποστέλλει τις αποκρίσεις του (αποτελέσματα μετρήσεων) και στην αμέσως επόμενη διαδοχικά ο client αποστέλλει αιτήματα - εντολές.
- `quality-level`: Πρόκειται για ρύθμιση της Unreal Engine. Η ποιότητα γραφικών αναφέρεται στον βαθμό λεπτομέρειας με τον οποίο η μηχανή γραφικών θα αποδίδει τα τρισδιάστατα μοντέλα και τις επιφάνειες αυτών. Όσο μεγαλύτερος αυτός, τόσο υψηλότερες φυσικά οι απαιτήσεις σε υπολογιστική ισχύ. Ο CARLA υποστηρίζει τις δύο ακραίες τιμές, "Low" και "Epic". Για τα πειράματα στην εργασία επελέγη η Epic.
- `ResX`, `ResY`: Η ανάλυση (διαστάσεις σε pixels) του παραθύρου όπου θα γίνεται το rendering των γραφικών. Και για αυτήν μεγαλύτερο μέγεθος απαιτεί υψηλότερη υπολογιστική ισχύ, επομένως όπως και η προηγούμενη, προσαρμόζεται αναλόγως με τις ικανότητες του εκάστοτε συστήματος. Στον υπολογιστή για την εργασία ετέθη ανάλυση 640x480.

Εικόνα 18: Το παράθυρο του CarlaUE4.

3.3.3 Clients

Στην προσομοίωση συμμετέχουν δύο clients. Ο ρόλος του πρώτου client στην διαδικασία είναι να λαμβάνει είσοδο από τον χρήστη (άνθρωπο ή αυτόνομο σύστημα) και να μεταβιβάζει τις εντολές αυτού στον server, ενώ του δεύτερου να είναι υπεύθυνος για την εκτέλεση του εκάστοτε σεναρίου. Η εκτέλεση του σεναρίου εμπεριέχει την ανάλυση του OpenSCENARIO, την διεκπεραίωση της δράσης επικοινωνώντας με τον server, την καταγραφή των γεγονότων (logging) στην γραμμή εντολών, και τον τερματισμό της προσομοίωσης.

Manual control client

Σε επίπεδο λογισμικού ο πρώτος client υλοποιείται από το script manual_control.py του scenario runner. Το script αυτό χρησιμοποιεί την PyGame ώστε να ανιχνεύει τα μέσα εισόδου από τον χρήστη, να λαμβάνει σήμα από αυτά, και να διοχετεύει αυτό στην προσομοίωση. Δημιουργεί ένα δεύτερο παράθυρο, το οποίο αντλώντας από την μηχανή γραφικών απεικονίζει το ίδιο γραφικό περιβάλλον με τον server και ουσιαστικώς είναι η διεπαφή του χρήστη με την προσομοίωση. Μέσω της εφαρμογής PyGame αυτής μπορεί ο χρήστης να ελέγχει το ego vehicle του σεναρίου είτε άμεσα με δική του είσοδο είτε ενεργοποιώντας τον αυτόματο πιλότο που διαθέτει ο CARLA.

Όπως και στον server, ο συγκεκριμένος client προϋποθέτει για να τρέξει τον καθορισμό ορισμένων παραμέτρων. Οι σημαντικότερες εξ αυτών είναι

- `map`: Όταν εκκινεί ο server, φορτώνει έναν προεπιλεγμένο χάρτη από την συλλογή χαρτών που διαθέτει στην βιβλιοθήκη γραφικών. Ο client μπορεί να προσδιορίσει μέσω meta-command διαφορετικό χάρτη για την προσομοίωση.
- `sync`: Προσδιορίζει ποιά διεργασία της προσομοίωσης (server ή client) θα δίνει τον κοινό παλμό (“tick”). Ο παλμός της προσομοίωσης είναι η ελάχιστη μονάδα χρόνου ανά την οποία επανυπολογίζονται όλες οι φυσικές τιμές και πραγματοποιούνται οι δράσεις. Όταν η προσομοίωση τρέχει σε sync mode, τότε η Unreal engine (το εκτελέσιμο C++) παράγει το επόμενο frame μόνο όταν πάρει εντολή από τον υπεύθυνο για το tick, που θα είναι η

αργότερη από τις συμμετέχουσες εφαρμογές, με την ανάλογη συνέπεια στην συχνότητα (ticking frequency). Αποδίδοντας στον client την ευθύνη για τον παλμό, επιτυγχάνεται η επαναληψιμότητα (repeatability) του σεναρίου, δηλαδή εξασφαλίζεται ότι θα προκύπτει το ίδιο αποτέλεσμα σε κάθε επανάληψη του ιδίου πειράματος υπό τις ίδιες συνθήκες. Αν η προσομοίωση δεν τρέχει σε sync mode, τότε η Unreal υπαγορεύει την συχνότητα παλμού και αξιοποιεί στο έπακρο τις δυνατότητες του υποκείμενου υλικού (hardware). Αυτό οδηγεί σε υψηλότερες επιδόσεις αλλά και στην μη επιθυμητή παρενέργεια η προσομοίωση να μην είναι αιτιοκρατική (deterministic), καθώς οι υπόλοιπες εφαρμογές δεν μπορούν να συμβαδίσουν με την συχνότητα παλμού.

- `timeout`: Η διαδικασία εκκίνησης του server διαρκεί κάποια δευτερόλεπτα. Συνεπώς, όταν server και client εκκινούν ταυτοχρόνως, μεσολαβεί ένα διάστημα προτού μπορεί να ανοίξει δίαυλος επικοινωνίας μεταξύ των. Η παράμετρος αυτή καθορίζει την μέγιστη χρονοκαθυστέρηση που θα αναμείνει ο client να λάβει σήμα από τον server προτού τερματίσει.
- Για το γραφικό παράθυρο του client προβλέπονται επίσης παράμετροι ανάλυσης και συχνότητας ανανέωσης εικόνας (frames per second - fps).

Εικόνα 19: Το παράθυρο του PyGame client στο ίδιο στιγμιότυπο με τον server.

Scenario runner client

Ο δεύτερος client στον κώδικα υλοποιείται εντός του αρχείου scenario_runner.py. Η λειτουργία του αφορά ειδικά τις περιπτώσεις χρήσης του προσομοιωτή όπου εκτελείται προκαθορισμένο σενάριο. Όταν η τιμή της παραμέτρου `CARLA_SCENARIO_RUNNER` στο driving_simulator.conf τίθεται True, τότε ο client ελέγχει την τιμή της παραμέτρου `CARLA_SCENARIO` και αναζητά το αντίστοιχο αρχείο στο προκαθορισμένο directory αποθήκευσης σεναρίων. Στην συνέχεια, στην περίπτωση που αφορά την εργασία, δηλαδή που το σενάριο είναι σε μορφή OpenSCENARIO XML, η συντακτική ανάλυση γίνεται σε πρώτη φάση από το script openscenario_configration.py, και έπειτα από το openscenario_parser.py.

Από το openscenario_configration.py ελέγχεται αρχικώς η συντακτική εγκυρότητα του αρχείου .xosc ως προς το γενικό πρότυπο XML και ειδικά το OpenSCENARIO XML, και στην συνέχεια εξάγονται οι βασικές πληροφορίες για το σενάριο:

- το όνομα αυτού,
- ο χάρτης στον οποίο διεξάγεται και
- οι συμμετέχουσες οντότητες με είδος (αυτοκίνητο ή πεζός) και συντεταγμένες αρχικής θέσης. Απαιτείται ένα τουλάχιστον όχημα να έχει την ιδιότητα του ego vehicle.

Οι πληροφορίες αυτές διαμορφώνουν τον προσομοιούμενο κόσμο μέσω meta-commands που αποστέλλει ο client για την αλλαγή του χάρτη και την δημιουργία και τοποθέτηση των οχημάτων. Έπειτα, το openscenario_parser.py είναι υπεύθυνο για τα υπόλοιπα στοιχεία του σεναρίου. Από αυτό εξάγεται, αν υπάρχει, `EnvironmentAction` για τον προσδιορισμό καιρικών συνθηκών, και επιστρέφεται κατάλληλο αντικείμενο `Weather`, όπως αυτό ορίζεται στο scenario_manager.py, για την εκτέλεση της meta-command. Επίσης εξάγονται οι `AssignControllerActions` και επιστρέφεται το όνομα του script του προσομοιωτή που καθορίζει το αρχείο ως μοντέλο ελέγχου. Ένα βασικό καθήκον του openscenario_parser.py είναι η ανάλυση όλων των `Maneuvers` και `Conditions` που περιέχει το σενάριο και η μετάφραση αυτών σε κατάλληλα αντικείμενα `AtomicBehavior` για να τα ερμηνεύσει ο προσομοιωτής. Τέλος, όλες οι συντεταγμένες που περιγράφονται στο σενάριο υφίστανται κατάλληλο μετασχηματισμό από τον προσανατολισμό δεξιάς χειρός που υποθέτουν τα πρότυπα του ASAM, σε προσανατολισμό αριστερής χειρός που υποθέτει η Unreal και κατά συνέπεια και ο CARLA.

Η προαναφερθείσα κλάση `AtomicBehavior` ορίζεται στο αρχείο atomic_behaviors.py. Περιγράφει αφηρημένα μία στοιχειώδη πράξη στο σενάριο, και λειτουργεί ως μητρική κλάση για όλες τις κλάσεις που περιγράφουν συγκεκριμένες πράξεις. Το openscenario_parser.py αναλύει κάθε ετικέτα του αρχείου και αντιστοιχίζει αυτές που περιγράφουν πράξεις στην κατάλληλη υπο-κλάση. Επομένως, από μία ετικέτα `EnvironmentAction` για τον καθορισμό καιρικών συνθηκών κατασκευάζει αντικείμενο κλάσης `ChangeWeather`, ενώ από ετικέτα `Maneuver` που περιέχει `SpeedAction` δημιουργείται αντικείμενο `ChangeActorTargetSpeed`, και ούτω καθ' εξής. Κάθε τέτοια κλάση περιέχει πεδία για όλες τις πληροφορίες εντός της ετικέτας OpenSCENARIO (ενδεικτικά η `ChangeActorTargetSpeed` αποθηκεύει όχημα αναφοράς, στόχο τελικής ταχύτητας, μέθοδο επίτευξης), αλλά και μέσω επεξεργασίας από το openscenario_parser, πληροφορίες για τις συνθήκες (`Condition`) έναρξης και τερματισμού. Όλες οι κλάσεις αυτές διαθέτουν συνάρτηση `update()`, η οποία εκτελείται σε κάθε παλμό της προσομοίωσης και πραγματοποιεί την εκάστοτε δράση.

Για τον έλεγχο της κατάστασης του σεναρίου από τις διάφορες συναρτήσεις που τρέχουν κατά την προσομοίωση, το scenario runner διαθέτει την κλάση `CarlaDataProvider`. Η κλάση ορίζεται στο αρχείο carla_data_provider.py και παρέχει ανά πάσα στιγμή πληροφορίες για τον κόσμο και κάθε επιμέρους οντότητα, όπως την θέση και την ταχύτητα αυτής.

Ο κώδικας του scenario runner επίσης διαθέτει ένα σύνολο έτοιμων μοντέλων ελέγχου (“actor controls”) για τις οντότητες. Ο εξωτερικός έλεγχος του ego vehicle από τον χρήστη υποστηρίζεται από το script external_control.py. Τα υπόλοιπα αυτοκίνητα της προσομοίωσης, τα NPCs, ελέγχονται από το npc_vehicle_control.py, στο οποίο ορίζεται συνάρτηση `run_step()` υπεύθυνη για την δράση του NPC σε κάθε παλμό. Όλη η εκτέλεση του σεναρίου οργανώνεται από το script scenario_manager.py, που καλεί διαδοχικά τα διάφορα μέρη του scenario runner ακολουθώντας τον παλμό της προσομοίωσης [31].

Στην περίπτωση που ο προσομοιωτής εκτελείται χωρίς σενάριο, τότε δεν χρησιμοποιείται ο κώδικας του scenario_runner, αλλά τον ρόλο του client αναλαμβάνει διαφορετικό αρχείο manual_control.py από το PythonAPI.

3.3.4 Shell scripts

Στην πράξη, η εκκίνηση της λειτουργίας του προσομοιωτή δεν γίνεται απ' ευθείας αλλά μέσω συμπλέγματος επικουρικών αρχείων κώδικα με απώτερο σκοπό την απλοποίηση της διαδικασίας στην τυπική περίπτωση χρήσης. Τελικώς για την πραγματοποίηση της προσομοίωσης εκτελείται από τον χρήστη το αρχείο shell start_simulator.sh, και για τον τερματισμό αυτής το stop_simulator.sh. Το start_simulator.sh αντλεί τιμές σταθερών καθορισμένων από τον χρήστη στο αρχείο διαμόρφωσης (configuration) driving_simulator.conf, και έπειτα καλεί το carla.sh. Το carla.sh είναι το κύριο αρχείο κώδικα που ουσιαστικώς ενορχηστρώνει την διαλειτουργικότητα μεταξύ server και client. Το stop_simulator.sh καλεί επίσης το carla.sh, θέτοντας την κατάλληλη μεταβλητή ώστε με την εκτέλεση να τερματιστεί το πρόγραμμα. Στο πλαίσιο της εργασίας η ενασχόληση με τον προσομοιωτή περιορίστηκε σε υψηλότερο επίπεδο (με εξαίρεση την εισαγωγή νέων χαρτών η οποία ήταν σε ακόμη χαμηλότερο επίπεδο). Συνεπώς δεν χρειάστηκαν να γίνουν αλλαγές ή συμπληρώσεις στα αρχεία αυτά, παρά μόνο στο driving_simulator.conf, το οποίο παρουσιάζεται στην συνέχεια.

Αρχείο διαμόρφωσης driving_simulator.conf

Στο αρχείο αυτό προσδιορίζονται οι τιμές παραμέτρων που αφορούν την λειτουργία του προσομοιωτή. Τέτοιες παράμετροι είναι η επιθυμητή διεύθυνση IP που θα τρέξει ο server, η ποιότητα γραφικών της unreal, η timeout του client, και πολλές άλλες που έχουν ήδη περιγραφεί. Ο χρήστης τις προσδιορίζει αυτές όχι απ' ευθείας καλώντας τα αντίστοιχα προγράμματα αλλά μέσω του αρχείου διαμόρφωσης. Ιδιαίτερης σημασίας για τους σκοπούς της εργασίας ήταν οι ακόλουθες παράμετροι:

- `CARLA_SCENARIO_RUNNER`: Αν λάβει τιμή True υποδεικνύει στον προσομοιωτή να χρησιμοποιήσει το υποσύστημα scenario runner για την εκτέλεση ενός σεναρίου.
- `CARLA_SCENARIO`: Το όνομα αρχείου OpenSCENARIO που περιέχει το σενάριο προς προσομοίωση. Μεταξύ διαφορετικών πειραμάτων στον ίδιο υπολογιστή αυτή είναι συνήθως η μοναδική παράμετρος που αλλάζει ο χρήστης.
- `CARLA_LOCAL_SIM_ONLY`: Αν λάβει τιμή True, το script start_simulator.sh εκκινεί μόνον τον server, και έπειτα η εκκίνηση των clients γίνεται από το run_standalone.sh. Αυτή επιλογή επιτρέπει στον server να προλάβει να αρχίσει να τρέχει προτού οι clients απαιτήσουν σύνδεση και χρειάζεται ειδικά εάν ο server τρέχει από σχετικά αργό δίσκο HDD.
- `CARLA_BIN_DIR`: Η τοποθεσία (directory) στον δίσκο του αρχείου binary του CARLA.
- `CARLA_SIM_DIR`: Το directory του υπόλοιπου κώδικα του προσομοιωτή (clients).
- `PYTHON_DIR`: Το directory εγκατάστασης της Python.

3.4 Το σχεδιαστικό πρόγραμμα RoadRunner

Το RoadRunner είναι το βασικό εργαλείο που χρησιμοποιήθηκε στην εργασία για τον σχεδιασμό οδικών δικτύων, την συγγραφή και δοκιμή πλήρων σεναρίων, και την μεταφορά αυτών στον προσομοιωτή CARLA. Αναπτύχθηκε από την εταιρεία Vector Zero και έχει εξαγορασθεί από την Math Works, η οποία συμμετέχει στον οργανισμό ASAM και κατά συνέπεια το πρόγραμμα υποστηρίζει πλήρως τα πρότυπα OpenDRIVE και OpenSCENARIO. Η έκδοση του λογισμικού που χρησιμοποιείται στην εργασία είναι η R2022b [19][32][33][34]. Αποτελείται από δύο κυρίως ημιαυτόνομα μέρη: το υπεύθυνο για την κατασκευή τρισδιάστατων γραφικών οδικών και αστικών δικτύων (“Scenes” στην ορολογία του λογισμικού), και το επονομαζόμενο RoadRunner Scenario, με το οποίο γίνεται η κατάστρωση των σεναρίων με τοποθέτηση οχημάτων και ανάθεση δράσεων στις Scenes αυτές. Αν και φαινομενικά τα δύο μέρη χρησιμοποιούνται ενιαία και στεγάζονται στην ίδια γραφική διεπαφή του προγράμματος, στην πραγματικότητα επιτελούν διακριτές λειτουργίες. Ως εκ τούτου, στην συνέχεια παρουσιάζονται χωριστά.

3.5.1 Σχεδιασμός χάρτη

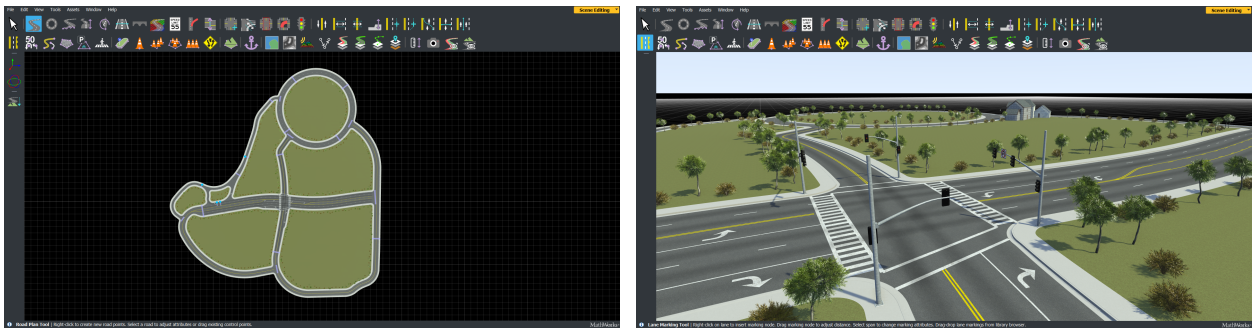
Ο Scene editor του RoadRunner παρέχει πολλά εργαλεία για τον λεπτομερή σχεδιασμό ενός οδικού δικτύου. Ο χρήστης μπορεί να τοποθετήσει στον χώρο έναν δρόμο, καθορίζοντας το μήκος αυτού, το πλήθος λωρίδων, το πλάτος και την κατεύθυνση κίνησης κάθε λωρίδας και το ύψος και πλάτος πεζοδρομίου. Παράλληλα, καθορίζει σημασιολογικά δεδομένα για κάθε οδό και λωρίδα, βάσει του προτύπου OpenDRIVE. Κάθε οδός έχει όριο ταχύτητος, και οι διάφορες λωρίδες μπορούν να έχουν διακριτούς ρόλους, όπως λεωφορειολωρίδα ή ΛΕΑ.

Από προεπιλογή, οι δρόμος είναι ευθύγραμμος, αλλά μπορεί να πάρει οποιοδήποτε σχήμα, είτε ακολουθώντας μαθηματική καμπύλη (όπως τόξο κύκλου), είτε συνδέοντας σύνολο σημείων που επιλέγει ο χρήστης στον χώρο. Το πρόγραμμα αναλαμβάνει αυτομάτως την κατάλληλη χάραξη διασταυρώσεων μεταξύ δύο ή περισσοτέρων δρόμων, υπό οποιαδήποτε γωνία. Ακόμη μία κατηγορία οδών που χαράσσεται αυτομάτως με την ευφυία του προγράμματος είναι οι λωρίδες εισόδου και εξόδου από κόμβους και αυτοκινητοδρόμους (slip lanes), επιτρέποντας στον χρήστη να επικεντρώνεται στις βασικές οδούς. Ο σχεδιασμός δρόμων με το RoadRunner δεν εξαντλείται στις δύο διαστάσεις, αλλά επεκτείνεται και στο ύψος: μπορούν δηλαδή να ανυψώνονται δρόμοι για την δημιουργία γεφυρών και ανισοπέδων διασταυρώσεων.

Μετά τον γεωμετρικό σχεδιασμό του δρόμου, το RoadRunner διαθέτει μεγάλη βιβλιοθήκη γραφικών για την επιφάνεια του δρόμου, με όλα τα είδη διαγραμμίσεων (διακεκομμένη, μονή συνεχής, διπλή συνεχής και πολλά άλλα), βέλη, διαβάσεις πεζών, αλλά και υφή και ποιότητα ασφάλτου. Η βιβλιοθήκη γραφικών επίσης περιέχει πλήθος οδικών πινακίδων και σημάτων, με όλα τα κυριότερα σήματα που απαντώνται σε κάθε δρόμο (όρια ταχύτητας, απαγορευτικά, STOP, υποχρεωτικές πορείες, σημάνσεις προτεραιότητας και λοιπά) και διαφόρων μεγεθών στύλους για φωτεινούς σηματοδότες. Μάλιστα, τα γραφικά στοιχεία αυτά περιέχονται σε διάφορες απεικονίσεις, ανάλογα με την χώρα προέλευσης (ΗΠΑ, Η.Β., Ευρώπη) ώστε να προσδίδεται περισσότερος ρεαλισμός ανά περίσταση.

Το RoadRunner τοποθετεί αυτομάτως διαχωριστικές νησίδες οπουδήποτε δημιουργούνται κενά μεταξύ δρόμων, όπως σε σημεία απόκλισης δύο λωρίδων ή σε διασταυρώσεις με μορφολογία που το απαιτεί. Φυσικά επιτρέπει στον χρήστη να επεξεργάζεται τις αυτομάτως δημιουργημένες νησίδες αλλά και να σχεδιάζει δικές του. Για τις διάφορες επιφάνειες, εκτός από παραλλαγές ασφάλτου και μπετόν, η βιβλιοθήκη γραφικών περιέχει και άλλες «υφές» (“textures”) για την πιστή αναπαράσταση του χώρου, όπως χώμα και χλόη. Οι χώροι στάθμευσης αντιμετωπίζονται ως διακριτά μέρη του οδικού δικτύου από το πρόγραμμα, και διατίθεται για αυτούς ειδική διαγράμμιση στην άσφαλτο.

Τέλος, η βιβλιοθήκη γραφικών του RoadRunner περιέχει μεγάλη ποικιλία μοντέλων αντικειμένων σε εύρος μεγεθών, που δεν είναι συστατικά του οδικού δικτύου αλλά είναι απαραίτητα για την διαμόρφωση του περιβάλλοντος χώρου. Τέτοια είναι η βλάστηση (δένδρα και θάμνα), τα κτήρια, οι δομές που συναντώνται συχνά σε πεζοδρόμια, όπως στάσεις λεωφορείων, στύλοι φωτισμού και πυροσβεστικοί κρουνοί, και πολλά διακοσμητικά στοιχεία. Μάλιστα, το πακέτο εγκατάστασης του RoadRunner περιέχει αρκετά παραδείγματα ολοκληρωμένων χαρτών (αρχεία .rrscene) που εξυπηρετούν ως πρότυπα. Η ευκολία χρήσης και υψηλή προσαρμοστικότητα του RoadRunner στην χάραξη δρόμων, σε συνδυασμό με την πληρότητα της βιβλιοθήκης γραφικών που διαθέτει, επιτρέπουν στον χρήστη να συνθέτει εξαιρετικά αληθοφανείς χάρτες [19][32][33][34].



Εικόνα 20: Απεικόνιση ενός χάρτη σε κάτοψη και προοπτική.

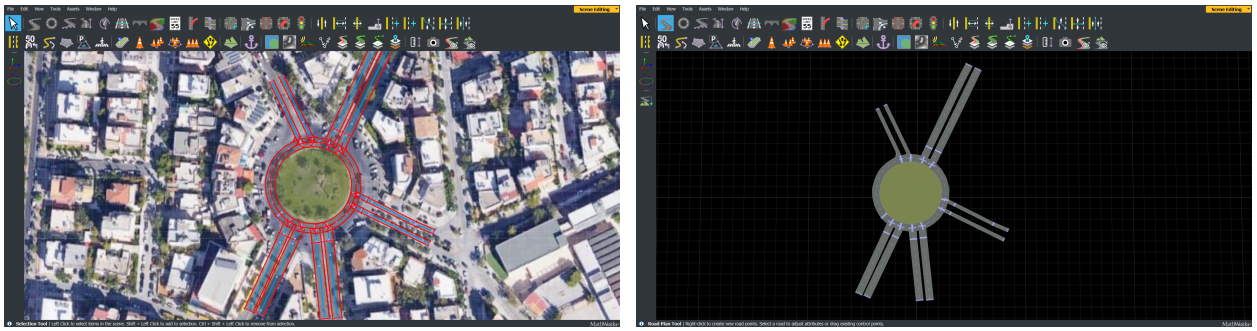
Κατασκευή χάρτη από πραγματικά δεδομένα

Εκτός από την χάραξη του οδικού δικτύου εκ του μηδενός με τα σχεδιαστικά εργαλεία, στο RoadRunner μπορούν να δημιουργηθούν χάρτες (scenes) και εισάγοντας κατάλληλα γεωγραφικά δεδομένα. Αυτό φυσικά είναι αναγκαίο για τις περιπτώσεις που θέλουμε να αναπαραστήσουμε στην προσομοίωση ένα πραγματικό οδικό δίκτυο.

Γεωγραφικά δεδομένα

Το RoadRunner υποστηρίζει αρκετούς τύπους δεδομένων GIS (geographic information system) όπως αεροφωτογραφίες, point cloud (γεωγραφικές συντεταγμένες των θέσεων των διαφόρων αντικειμένων) και δεδομένα υψομέτρου (elevation) για κάθε σημείο του χώρου. Συνδυάζοντας αυτά, μπορεί ο χρήστης να χαράξει τους δρόμους χρησιμοποιώντας ως πρότυπο την κάτοψη του χώρου από την φωτογραφία και το πρόγραμμα θα τους προσαρμόσει αυτομάτως στην τοπολογία του εδάφους. Τελικώς ο χάρτης που θα προκύψει θα έχει υψηλή πιστότητα χάρη στα δεδομένα γεωγραφικών συντεταγμένων. Η μέθοδος αυτή δεν χρησιμοποιήθηκε για κατασκευή χαρτών στο πλαίσιο της εργασίας, γιατί αφ' ενός δεν

χρειάστηκε τόση ακρίβεια για τα σενάρια στα οποία έγιναν πειράματα, και αφ' ετέρου οι απαραίτητες γεωγραφικές πληροφορίες, πλην των δορυφορικών φωτογραφιών, δεν είναι διαθέσιμες για την Ελλάδα.



Εικόνα 21: Σχεδίαση χάρτη με πρότυπο δορυφορική φωτογραφία.

Η δορυφορική φωτογραφία που χρησιμοποιείται ως πρότυπο στην εικόνα 21 για τον σχεδιασμό του χάρτη είναι βοηθητική· δεν γίνεται μέρος της σκηνής και κατά συνέπεια ούτε των σεναρίων επάνω σε αυτήν.

OpenStreetMap

Για την αποτύπωση αληθινών αστικών περιοχών στο RoadRunner απεδείχθη καταλληλότερη η εισαγωγή δεδομένων OpenStreetMap (osm). Ο οργανισμός OpenStreetMap Foundation παρέχει γεωγραφικά δεδομένα, συμπεριλαμβανομένου χάρτη ολόκληρης της Γης με πλήρεις οδικές πληροφορίες. Από την ιστοσελίδα openstreetmap.org μπορούν να ληφθούν δωρεάν αυτά τα δεδομένα για οποιαδήποτε περιοχή, κωδικοποιημένα σε XML στον τύπο αρχείου osm. Μέσα στο αρχείο περιγράφεται πλήρως με κατάλληλη αναπαράσταση η επιλεγμένη περιοχή του χάρτη: κάθε οδός αναλύεται σε σειρά πολλών σημειακών κόμβων, των οποίων οι συντεταγμένες προσδιορίζονται ακριβώς. Επίσης συμπεριλαμβάνονται πληροφορίες για τα όρια ταχύτητος, τα πλάτη των δρόμων και την κατεύθυνση κάθε λωρίδας [35].

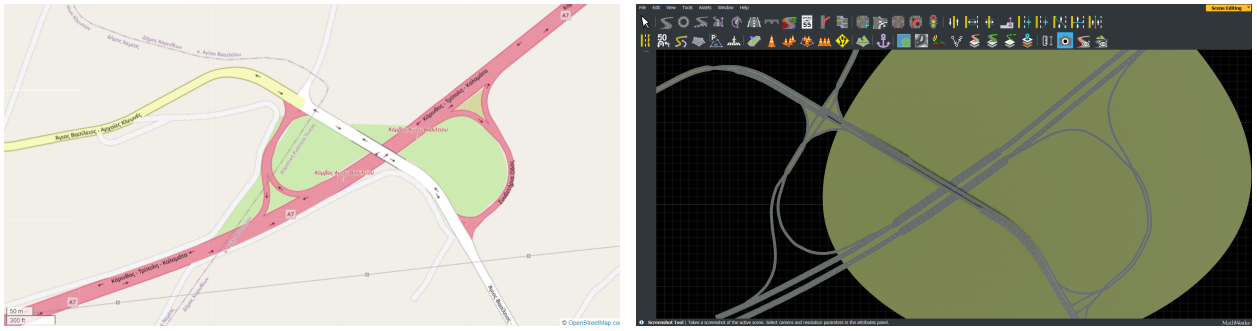
Το RoadRunner αντλεί από το αρχείο .osm τις πληροφορίες αυτές και αποτυπώνει το οδικό δίκτυο. Στην συνέχεια, ο χρήστης μπορεί να επέμβει και να κάνει διορθώσεις, καθώς υπάρχει περίπτωση το αρχείο .osm να περιέχει ανακρίβειες. Τέτοιες παρατηρήθηκαν στα πλάτη των λωρίδων, και στην αντιμετώπιση μονοπατιών για πεζούς ως δρόμων αυτοκινήτων. Επιπλέον σε ορισμένους αυτοκινητοδρόμους δεν ήταν καταγεγραμμένη η ΛΕΑ. Ωστόσο, οι αβελτηρίες αυτές του .osm είναι δευτερευούσης σημασίας εν σχέσει με το βασικό προτέρημα, που είναι η ακριβής περιγραφή του σχήματος και μήκους κάθε δρόμου στον χάρτη.

Στο Παράρτημα Α’ παρουσιάζεται η κατασκευή του (υποτυπώδους) χάρτη του campus του Πολυτεχνείου στο RoadRunner με χρήση OpenStreetMap.

```
66  <node id="250691671" visible="true" lat="37.9767209" lon="23.7343863">
67   <tag k="crossing" v="traffic_signals"/>
68   <tag k="highway" v="crossing"/>
69  </node>
70  <node id="250691679" visible="true" lat="37.9766165" lon="23.7344574"/>
71  <node id="250691681" visible="true" lat="37.9765445" lon="23.7344673"/>
72  <node id="250691695" visible="true" lat="37.9772234" lon="23.7356124"/>
```

```
26183  <way id="789798944" visible="true" version="3" changeset="128752346"
26184   <nd ref="250691681"/>
26185   <nd ref="8542629904"/>
26186   <nd ref="954712452"/>
26187   <nd ref="954712457"/>
26188   <nd ref="42240105"/>
26189   <tag k="foot" v="yes"/>
26190   <tag k="highway" v="primary"/>
26191   <tag k="int_name" v="Stadiou"/>
26192   <tag k="lanes" v="3"/>
26193   <tag k="lit" v="yes"/>
26194   <tag k="name" v="Σταδίου"/>
26195   <tag k="name:es" v="Stadiou"/>
26196   <tag k="oneway" v="yes"/>
26197   <tag k="ref" v="ΕΟ1"/>
26198   <tag k="surface" v="asphalt"/>
26199   <tag k="trolley_wire" v="yes"/>
26200  </way>
```

Εικόνα 22: Δείγμα αρχείου .osm: ο δρόμος περιγράφεται από ακολουθία κόμβων (nodes) που αναπαριστούν σημεία με συγκεκριμένο πλάτος και μήκος.

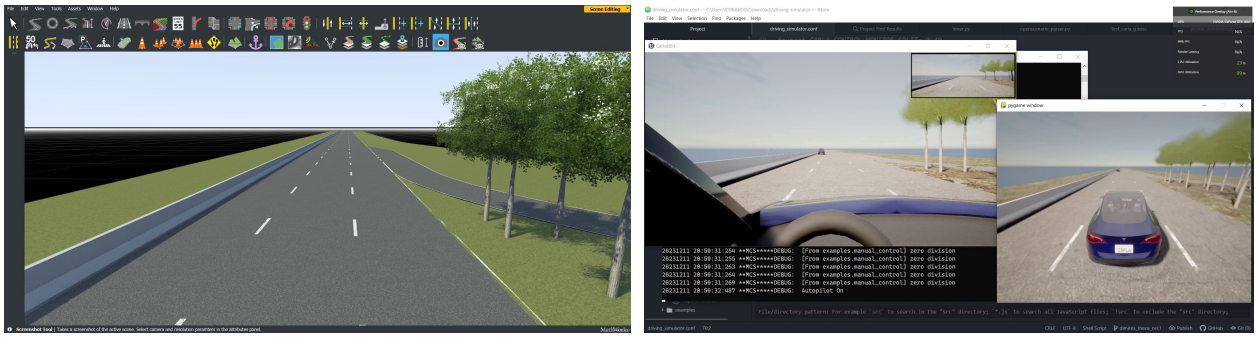


Εικόνα 23: Κόμβος της ΕΟ7 στο OpenStreetMap και αποτύπωση αυτού στο RoadRunner.

Εξαγωγή

Το RoadRunner μπορεί να παραγάγει από έναν χάρτη (αρχείο .rrscense) το αντίστοιχο αρχείο OpenDRIVE. Όμως παρέχει και πολλές άλλες επιλογές εξαγωγής (export), συμπεριλαμβανομένης και ειδικής επιλογής για τον προσομοιωτή CARLA. Αυτή δημιουργεί εκτός από το αρχείο .xodr και αρχείο Filmbox (.fbx), που χρησιμοποιείται για την μεταφορά δεδομένων τρισδιάστατων γραφικών μεταξύ διαφορετικών εφαρμογών, και αρχείο .geojson που περιέχει γεωγραφικά δεδομένα. Οι επιπρόσθετες πληροφορίες σε αυτά τα αρχεία είναι αναγκαίες για την μεταφορά και κατασκευή του χάρτη στον προσομοιωτή, καθώς το αρχείο OpenDRIVE περιγράφει κυρίως τα δομικά στοιχεία του οδικού δικτύου και όχι τον περιβάλλοντα χώρο. Τα μοντέλα της βιβλιοθήκης γραφικών του RoadRunner είναι συμβατά με την Unreal, συνεπώς μεταφέρονται και εμφανίζονται κανονικά στον CARLA. Η διαδικασία

αυτή, εξαγωγής χάρτη από το RoadRunner και ανακατασκευής του στην Unreal Engine παρουσιάζεται αναλυτικά στο Παράρτημα Β’.



Εικόνα 24: Ο ίδιος χάρτης στο RoadRunner και στον CARLA εν μέσω προσομοίωσης.

3.5.2 Συγγραφή σεναρίου

Το RoadRunner Scenario από άποψη λογισμικού λειτουργεί ως πρόσθετο στο κυρίως πρόγραμμα, και ο ρόλος του είναι η συγγραφή σεναρίων βασισμένων στις Scenes που έχουν κατασκευαστεί με το RoadRunner. Ενώ στο RoadRunner γίνεται η επεξεργασία του χάρτη, στο RoadRunner Scenario, όπου ο ίδιος χάρτης αποτελεί το στατικό μέρος του σεναρίου, γίνεται η επεξεργασία των δυναμικών μερών, δηλαδή ο καθορισμός των κινουμένων σωμάτων και ο προσδιορισμός της συμπεριφοράς αυτών. Το πρόγραμμα παρέχει γραφικό εργαλείο για την διάταξη των διαφόρων actors στον χώρο και τον οπτικό προγραμματισμό των κινήσεών τους (visual programming), και επίσης περιέχει ενσωματωμένο προσομοιωτή για την εκτέλεση και δοκιμή των σεναρίων. Η εναλλαγή μεταξύ των δύο προγραμμάτων γίνεται στιγμιαία εντός της ίδιας γραφικής διεπαφής, επιτρέποντας στον χρήστη να εργάζεται παράλληλα στα δύο σκέλη του σεναρίου.

Στην ορολογία του RoadRunner, τα κινούμενα σώματα στο σενάριο ονομάζονται “actors”. Όλοι οι actors των σεναρίων της εργασίας είναι αυτοκίνητα. Το RoadRunner Scenario διαθέτει έτοιμα τρισδιάστατα μοντέλα διαφόρων τύπων οχημάτων, από μικρό αυτοκίνητο πόλης έως φορτηγό. Ένα όχημα εισάγεται στον χάρτη σύροντας το αντίστοιχο asset από το παράθυρο της βιβλιοθήκης γραφικών. Κάθε όχημα που τοποθετείται, προστίθεται αυτομάτως στο σενάριο ως διακριτή οντότητα, και ο χρήστης μπορεί να επεξεργαστεί το χρώμα αυτού, το μοναδικό όνομά του ως actor με το οποίο αναφέρεται και στο αρχείο OpenSCENARIO, και την ακριβή θέση του στον χώρο.

Η θέση κάθε αυτοκινήτου στον χάρτη προσδιορίζεται ως απόσταση (σε δύο διαστάσεις) από ένα σημείο αναφοράς (anchor). Αν όλα τα αυτοκίνητα έχουν τοποθετηθεί χρησιμοποιώντας την ίδια anchor, τότε μετακινώντας αυτήν μπορεί να μεταφερθεί αυτούσια η συστοιχία οχημάτων σε διαφορετική περιοχή του χάρτη, και να εκτελεστεί το ίδιο σενάριο με διαφορετικές αρχικές θέσεις. Στις περισσότερες περιπτώσεις, όταν το σενάριο μετατραπεί από RoadRunner σε αναπαράσταση OpenSCENARIO, οι θέσεις των οχημάτων δεν προσδιορίζονται με απόλυτες συντεταγμένες, αλλά χρησιμοποιώντας ως αναφορά τα δομικά στοιχεία των οδών του χάρτη, ήτοι τους δρόμους και τις λωρίδες. Αυτά φέρουν μοναδικά

αναγνωριστικά σύμφωνα με το πρότυπο OpenDRIVE, βάσει του οποίου έχει κατασκευαστεί η scene στο RoadRunnner.

Για τον προγραμματισμό των κινήσεων και γενικώς της συμπεριφοράς των οχημάτων κατά την εκτέλεση του σεναρίου, το πρόγραμμα διαθέτει ειδικό εργαλείο οπτικού προγραμματισμού ("Logic editor"), οιονεί διάγραμμα ροής. Σε κάθε όχημα του σεναρίου αντιστοιχεί μία αλυσίδα δράσεων (action phases). Μία δράση αναπαριστά μία ολοκληρωμένη αλλά διακριτή πράξη του αυτοκινήτου, που μπορεί να απαρτίζεται από σειρά στοιχειωδών πράξεων. Συγκεκριμένα, κάθε action phase καθορίζεται αρχικώς από το είδος και έπειτα από τις ανάλογες παραμέτρους.

Τα κυριότερα είδη είναι:

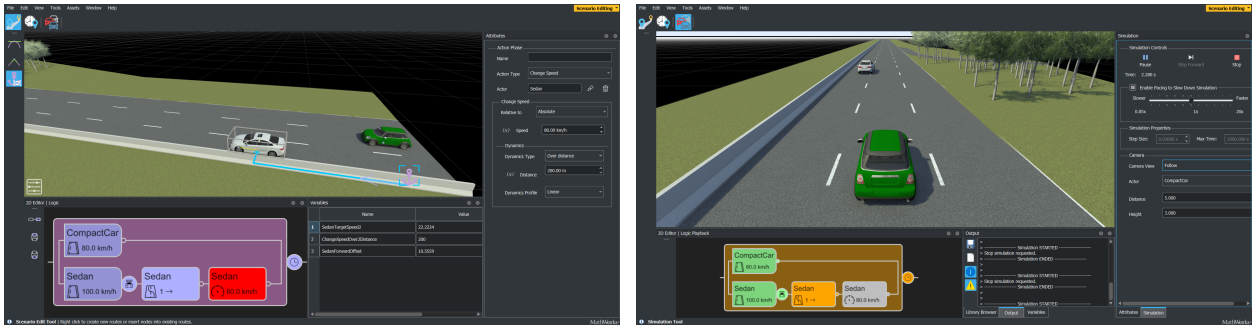
- Change Speed
- Change Lane
- Change Lateral Offset
- Change Longitudinal Distance
- Follow Path

Η Change Speed, παραδείγματος χάριν, περιγράφει την επιτάχυνση του αυτοκινήτου στο οποίο αντιστοιχεί, προσδιορίζοντας τελική ταχύτητα, σύστημα αναφοράς για την μέτρηση της ταχύτητας (απόλυτη ή σχετική με άλλον actor), dynamics type και dynamics profile κατά το πρότυπο OpenSCENARIO. Στην πραγματικότητα κάθε action phase αντιστοιχίζεται μονοσήμαντα στην ισοδύναμη Action του OpenSCENARIO. Επομένως, το προηγούμενο παράδειγμα επί της ουσίας είναι μία `<SpeedAction>` του OpenSCENARIO με τις παραμέτρους να προσδιορίζουν τα αντίστοιχα πεδία της ετικέτας, που περιγράφονται στο Κεφάλαιο 2.

Κάθε δράση χρειάζεται κάποιο χρονικό διάστημα για να πραγματοποιηθεί από το αυτοκίνητο στην προσομοίωση. Αφ' ης στιγμής ολοκληρωθεί, το λογικό διάγραμμα μεταβαίνει αυτομάτως στην επόμενη φάση. Μεταξύ των φάσεων όμως μπορούν να προστεθούν επιπλέον συνθήκες που πρέπει να πληρούνται για να γίνει η μετάβαση. Αυτές αντιστοιχούν στις Conditions του OpenSCENARIO.

Η προεπιλεγμένη συμπεριφορά για ένα αυτοκίνητο στο οποίο έχει προσδοθεί ταχύτητα μέσω του Logic editor, είναι να κινείται με αυτήν την σταθερή ταχύτητα (δεδομένου ότι δεν ακολουθεί άλλη action phase στο διάγραμμά του) στον ίδιο δρόμο που έχει τοποθετηθεί και παραμένοντας στην λωρίδα όπου βρίσκεται. Για την χάραξη τροχιάς που δεν προκύπτει από αυτόν τον κανόνα, αλλά και γενικώς για την πλοήγηση του αυτοκινήτου σε οποιοδήποτε σημείο του χάρτη και την πραγματοποίηση οποιουδήποτε ελιγμού, το RoadRunner Scenario υποστηρίζει λεπτομερή σχεδιασμό διαδρομής (path following). Η τροχιά του αυτοκινήτου προκύπτει συνδέοντας σειρά σημείων αναφοράς που καθορίζει ο χρήστης. Στην συνέχεια μπορεί για κάθε σημείο αναφοράς και για κάθε διάστημα μεταξύ σημείων να προσδιορίσει τα φυσικά μεγέθη της κίνησης του αυτοκινήτου για να ελέγξει την ταχύτητα αυτού σε κάθε στιγμή.

Κάθε ιδιότητα (attribute) του σεναρίου, όπως οι παράμετροι των action phases και η σχετική απόσταση θέσης αυτοκινήτου από anchor (offset), μπορεί να συνδεθεί με μία μεταβλητή (variable). Το RoadRunner Scenario διατηρεί κατάλογο των μεταβλητών και επιτρέπει την αλλαγή των τιμών αυτών, καθιστώντας εφικτή την προγραμματιστική επεξεργασία του σεναρίου με εξωτερικό εργαλείο μέσω του gRPC API του RoadRunner [19][32][33][34].



Εικόνα 25: Αριστερά κατάστρωση σεναρίου με τον logic editor και μεταβλητές. Δεξιά δοκιμή του σεναρίου με τον ενσωματωμένο προσομοιωτή.

Τέλος, από το αρχείο του σεναρίου RoadRunner .rrscenario μπορεί να παραχθεί (export) το αντίστοιχο OpenSCENARIO XML (αρχείο .xosc), το οποίο θα χρησιμοποιεί αναφορές στον χάρτη βάσει των αναγνωριστικών στην αναπαράσταση OpenDRIVE (αρχείο .xodr) αυτού.

Remote Procedure Calls (RPC)

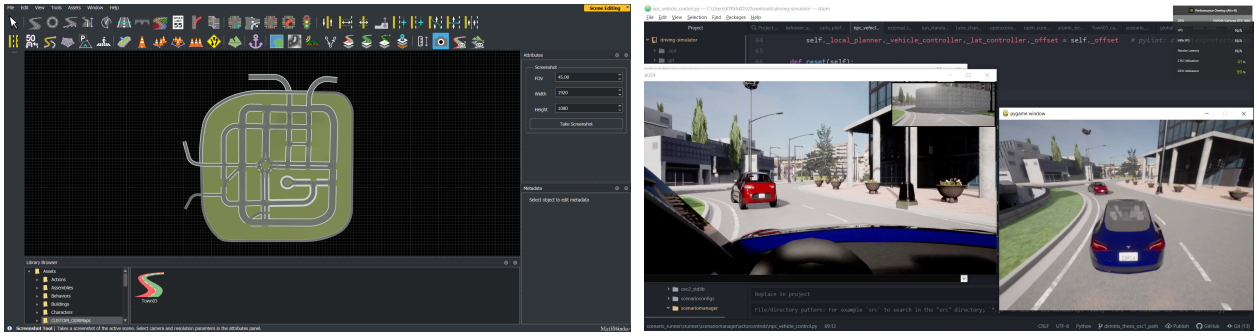
Το RoadRunner παρέχει την δυνατότητα προγραμματιστικού ελέγχου της διεπαφής (UI) του μέσω Remote Procedure Calls (RPC). Η ιδέα της «εξ αποστάσεως κλήσης διαδικασίας» είναι να μπορεί μία διαδικασία (procedure) ενός προγράμματος (διεργασία - process) που εκτελείται σε κάποιον υπολογιστή να καλεί μίαν άλλη διαδικασία ενός άλλου προγράμματος εκτελούμενου από άλλον ή τον ίδιο υπολογιστή. Αποτέλεσμα της κλήσης αναμένεται να είναι η επιτέλεση της διαδικασίας σαν να ήταν τοπική [36].

Το gRPC είναι ένα framework ανοικτού κώδικα που υλοποιεί το πρωτόκολλο RPC με αρχιτεκτονική client-server. Είναι ιδιαιτέρως εύχρηστο καθώς διατίθεται σε πολλές γλώσσες προγραμματισμού [37]. Το RoadRunner διαθέτει ενσωματωμένη προγραμματιστική διεπαφή (API) για το gRPC, η οποία διαμεσολαβεί μεταξύ του ίδιου του προγράμματος RoadRunner, που έχει τον ρόλο του server, και του προγράμματος του χρήστη που αποστέλλει κλήσεις εξ αποστάσεως, το οποίο παίζει τον ρόλο του client. Μέσω της διεπαφής, από ένα script Python μπορούν να κληθούν διαδικασίες του RoadRunner για την επεξεργασία των τιμών των προκαθορισμένων μεταβλητών που αντιστοιχούν σε παραμέτρους του σεναρίου, και για την εξαγωγή του σεναρίου σε μορφή OpenSCENARIO [34][38]. Η δυνατότητα αυτή είναι ιδιαιτέρως χρήσιμη για την παραγωγή πολλαπλών παραλλαγών σεναρίου με διαφορετικές τιμές παραμέτρων, σε πολύ λίγο χρόνο.

Συγγραφή σεναρίου για προϋπάρχοντα χάρτη του CARLA

Το RoadRunner υποστηρίζει και την αντίστροφη διαδικασία από την εξαγωγή, δηλαδή εισαγωγή (import) αρχείου OpenDRIVE και κατασκευή χάρτη εντός του προγράμματος. Στον κώδικα του CARLA παρέχονται τα αρχεία .xodr των προκατασκευασμένων πόλεων που περιέχει ο server του προσομοιωτή. Επομένως θεωρητικά είναι δυνατή η εποπτεία ενός χάρτη του CARLA και η συγγραφή σεναρίων σε αυτόν μέσω RoadRunner. Βέβαια, δεν γίνεται να αναπαρασταθούν στο δεύτερο τρισδιάστατα μοντέλα του πρώτου, επομένως ο χάρτης στο RoadRunner δεν περιέχει κτήρια και λοιπά στοιχεία πέραν του οδικού δικτύου. Ένα σημαντικότερο πρόβλημα είναι ότι το αρχείο .xodr δεν περιέχει την πληροφορία για το

υψόμετρο κάθε σημείου, με αποτέλεσμα να αποτυπώνεται ισόπεδος. Όταν το σενάριο μεταφέρεται στον CARLA, όλα τα οχήματα έχουν την ίδια σχεδόν μηδενική τιμή στην συντεταγμένη z (κατακόρυφο άξονα), με συνέπεια αν το σημείο του δρόμου που έχουν τοποθετηθεί βρίσκεται σε μεγαλύτερο ύψος να αποτυγχάνει το rendering του οχήματος. Τέλος, το αρχείο OpenDRIVE του CARLA δεν χρησιμοποιεί την σύμβαση του RoadRunner με την ανάθεση αναγνωριστικών σε κάθε δρόμο για την περιγραφή της θέσης των οχημάτων, οπότε απαιτείται προσδιορισμός αυτής με απόλυτες συντεταγμένες.



Εικόνα 26: Η Town03 στον RoadRunner και μεταφορά σεναρίου από εκεί στον CARLA.

Στο πλαίσιο της εργασίας εισήχθη στο RoadRunner ο χάρτης της Town03 του CARLA και σχεδιάστηκε ένα απλό σενάριο το οποίο ύστερα από αρκετές τροποποιήσεις εκτελέσθηκε από τον CARLA, ωστόσο οι ανωτέρω ασυμβατότητες καθιστούν την μέθοδο πρακτικώς μη εφαρμόσιμη για σύνθετα σενάρια.

3.5 Υπολογιστικό σύστημα

Η ανάπτυξη όλης της εργασίας στηρίχθηκε σε φορητό υπολογιστή (laptop) με επεξεργαστή Intel i7-9750H, 16GB RAM, κάρτα γραφικών NVIDIA GTX1650, δίσκους SSD και HDD και λειτουργικό σύστημα Windows 10 64bit. Ορισμένα τμήματα του προσομοιωτή εγκαταστάθηκαν στον δίσκο SSD του υπολογιστή, αλλά λόγω μεγάλου μεγέθους τα υπόλοιπα (Unreal engine και CARLA build) έπρεπε να εγκατασταθούν στον HDD. Με τις κατάλληλες ρυθμίσεις επετεύχθη ικανοποιητικά ομαλή ροή εκτέλεσης (gameplay).

3.6 Περιορισμοί

Προτού παρουσιασθούν αναλυτικά τα κατασκευασμένα για την εργασία σενάρια, πρέπει να διευκρινισθούν οι περιορισμοί που επεβλήθησαν στα σενάρια αυτά ως αποτέλεσμα ασυμβατοτήτων μεταξύ παραγόμενου από το RoadRunner σεναρίου σε μορφή ASAM OpenSCENARIO και CARLA scenario runner. Την περίοδο διεξαγωγής των πειραμάτων της εργασίας, δεν υποστηρίζονταν ακόμη από το scenario runner τα ακόλουθα στοιχεία του προτύπου:

- Η τιμή "runningState" για το πεδίο state ετικέτας `<StoryboardElementStateCondition>`. Στο Κεφάλαιο 2 εξηγείται η γενική λειτουργία των Conditions στο πρότυπο. Η συγκεκριμένη κατηγορία Condition βασίζεται στην κατάσταση (state) ενός Element του σεναρίου, και συγκεκριμένα ενός Event. Στην πράξη, η συνθήκη αυτή εφαρμόζεται στο πρώτο Event ολόκληρου του σεναρίου, και χρησιμοποιείται για την πυροδότηση της ακολουθίας ενεργειών. Όταν το πρώτο Event τεθεί σε κατάσταση runningState, αυτό σημαίνει την έναρξη της εκτέλεσης του σεναρίου από τον προσομοιωτή. Η μη υποστήριξη της τιμής αυτής από το scenario runner ήταν σοβαρή έλλειψη, ωστόσο βρέθηκε λύση παρατηρώντας τα σενάρια δημοσιευμένα από την ίδια την ομάδα ανάπτυξης του CARLA. Αυτά παρακάμπτουν την ανάγκη χρήσης της runningState θέτοντας ως συνθήκη έναρξης του σεναρίου ένα χαμηλό κατώφλι απόστασης που έχει διανυθεί από το ego vehicle, με μία `<RelativeDistanceCondition>`. Με αυτόν τον τρόπο, μόλις το όχημα που ελέγχει ο χρήστης αρχίσει να ολισθαίνει, το σενάριο εκκινείται.
- Προκειμένου να αναγνωρισθεί το ego vehicle του σεναρίου ως αυτό, πρέπει το ScenarioObject που το αναπαριστά να έχει στο γνώρισμα name την τιμή "hero". Πέραν αυτής, οι περισσότερες προδιαγραφές για τις οντότητες στο τμήμα ορισμών Entities δεν λαμβάνονται υπ' όψιν. Συγκεκριμένα, το scenario runner επιλέγει το κατάλληλο μοντέλο 3d από την βιβλιοθήκη του CARLA, βάσει του πεδίου name της κάθε ετικέτας `<Vehicle>`. Αυτό σημαίνει ότι, αφ' ενός πρέπει στο σενάριο που προκύπτει από το RoadRunner να καθορίζονται τα ονόματα των επιθυμητών οχημάτων βάσει της ονοματοδοσίας των μοντέλων του CARLA, και αφ' ετέρου ότι οι υπόλοιπες πληροφορίες εντός της ετικέτας αγνοούνται, καθώς προκύπτουν από τα εγγενή στοιχεία του εκάστοτε μοντέλου. Αυτές οι πληροφορίες είναι οι επιδόσεις του οχήματος, εντός της ετικέτας Performance, οι διαστάσεις (τα «σύνορα») του μοντέλου ως γεωμετρικού στερεού στον χώρο εντός της ετικέτας BoundingBox, και τεχνικά χαρακτηριστικά των αξόνων εντός της `<Axles>`. Αυτή η συμπεριφορά του scenario runner βέβαια είναι επιθυμητή, καθώς στην πράξη τελικώς τα σενάρια εκτελούνται στον προσομοιωτή CARLA με τα οχήματα που αυτός διαθέτει, ωστόσο εν γένει θέτει έναν ακόμη περιορισμό στην δυνατότητα παραμετροποίησης του σεναρίου μέσω του προτύπου.
- Στις συνθήκες που αφορούν απόσταση, όπως η προαναφερθείσα RelativeDistanceCondition, το πεδίο freespace καθορίζει, αναλόγως αν λάβει τιμή "false" ή "true", αν μία απόσταση που μετρείται εν σχέσει με μία Entity θα μετρηθεί απλώς από το σημείο αναφοράς αυτής, ή θα λάβει υπ' όψιν και τα bounding boxes. Για το πεδίο αυτό δεν υποστηρίζεται η τιμή "true". Ωστόσο, αυτή η έλλειψη δεν προκάλεσε σοβαρά προβλήματα, καθώς τα σενάρια περιέχουν συμβατικά αυτοκίνητα Ι.Χ., όπου η διαφορά μεταξύ των δύο μετρήσεων είναι μικρή.
- Στις Actions επιτάχυνσης αυτοκινήτου (`<SpeedAction>` εντός `<LongitudinalAction>`), η ετικέτα `<SpeedActionDynamics>` καθορίζει τα φυσικά μεγέθη της κίνησης που θα

επιτελεστεί. Εντός της, το πεδίο dynamicsDimension δίνει στον συγγραφέα του σεναρίου την επιλογή του μεγέθους που θα έχει υπό έλεγχο στην κίνηση αυτήν. Συγκεκριμένα, λαμβάνει τις τιμές “rate” (επιτάχυνση), “time” (χρόνος) και “distance” (μετατόπιση). Συνεπώς ο συγγραφέας καθορίζει είτε με πόση (σταθερή) επιτάχυνση θα γίνει η αλλαγή ταχύτητας, είτε σε πόσο χρόνο θα φθάσει η τιμή της ταχύτητας του αυτοκινήτου στην επιθυμητή, είτε πόση απόσταση θα έχει διανύσει το αυτοκίνητο έως ότου γίνει αυτό. Σε κάθε περίπτωση, τα υπόλοιπα δύο μεγέθη τα χειρίζεται καταλλήλως ο προσομοιωτής.

- Το scenario runner δεν υποστηρίζει καθόλου την επιλογή για επιτάχυνση (“rate”). Θεωρητικά η συμπλήρωση αυτής στον κώδικα του προσομοιωτή θα ήταν εφικτή, αφού ο CARLA παρέχει API με εντολές ελέγχου της κίνησης του οχήματος σε επίπεδο δυναμικής. Ωστόσο δεν κρίθηκε απαραίτητη, διότι η ομαλή επιτάχυνση δεν ανταποκρίνεται στην πραγματικότητα (η ταχύτητα των αυτοκινήτων δεν αυξάνεται γραμμικά καθ’ όλην την διάρκεια της επιτάχυνσης [39]) και συνεπώς δεν θα ήταν σκόπιμη η εφαρμογή της σε σενάρια. Αντιθέτως, ορίζοντας χρόνο ή απόσταση, μεγέθη τα οποία αντιλαμβάνεται και χειρίζεται καλύτερα και ο άνθρωπος οδηγός, ο προσομοιωτής αναλαμβάνει να διαμορφώσει την καμπύλη ταχύτητας του οχήματος στο αντίστοιχο διάστημα. Υποθέτοντας ότι ο προσομοιωτής έχει υλοποιηθεί με προσοχή στην επιστημονική πιστότητα ώστε να χρησιμοποιεί κατάλληλα την μηχανή φυσικής, η κίνηση που θα παραγάγει τότε θα προσομοιάζει καλύτερα την πραγματική.
- Όμως, και στην περίπτωση που επιλέγεται για την dynamicsDimension τιμή “time” ή “distance”, υπάρχει περιορισμός. Αυτός έγκειται στις μη υποστηριζόμενες από το scenario runner τιμές του πεδίου dynamicsShape. Το πεδίο αυτό καθορίζει το είδος καμπύλης που θα ακολουθεί η μεταβολή του μεταβαλλόμενου μεγέθους (εν προκειμένω ταχύτητα). Οι δυνατές τιμές είναι “cubic” (πολυωνυμική τρίτου βαθμού), “linear” (γραμμική), “sinusoidal” (ημιτονοειδής) και “step” (βηματική). Παρ' ότι το scenario runner αποδέχεται μόνο την linear, στην πραγματικότητα στο εκτελούμενο από τον CARLA scenario η αλλαγή της ταχύτητας δεν πραγματοποιείται γραμμικά, δηλαδή με σταθερό ρυθμό. Αυτό είναι από τα κυριότερα σημεία ασυμφωνίας μεταξύ προτύπου και προσομοιωτή.
- Τέλος, μία ιδιαιτερότητα του scenario runner που δεν ανάγεται σε ελλιπή υποστήριξη είναι ότι πρέπει στις περισσότερες περιπτώσεις τα οχήματα στο σενάριο να ξεκινούν από ηρεμία (μηδενική ταχύτητα), επειδή αν τους έχει προσδοθεί αρχική ταχύτητα τότε αρχίζουν να κινούνται προτού φορτώσει πλήρως το παράθυρο PyGame ώστε να έχει έλεγχο ο χρήστης.

Κεφάλαιο 4: Υλοποίηση

Αρχικώς, για την συγγραφή των σεναρίων έχουν γίνει οι εξής υποθέσεις:

- Σε κάθε εκτέλεση του σεναρίου ένα μοναδικό και συγκεκριμένο όχημα είναι το επίκεντρο της προσομοίωσης (ego). Το όχημα αυτό ελέγχεται από τον χρήστη. Μεταξύ διαφορετικών προσομοιώσεων του ιδίου σεναρίου, μπορεί ο ρόλος του ego να εναλλάσσεται μεταξύ των συμμετεχόντων οχημάτων.
- Πλευρά οδήγησης η δεξιά.

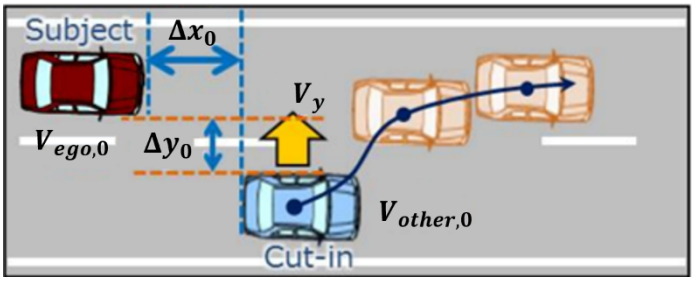
Ακολουθεί η περιγραφή των σεναρίων της εργασίας και της διαδικασίας σχεδιασμού και εκτέλεσης αυτών.

4.1 Cut-in σε αυτοκινητόδρομο

4.1.1 Περιγραφή

Ο EuroNCAP ορίζει ως "cut-in" («αποκοπή») την απότομη κίνηση αυτοκινήτου από την λωρίδα που οδηγεί σε παρακείμενη ίδιας κατεύθυνσης, στην οποία κινείται σε μικρή απόσταση πίσω άλλο αυτοκίνητο και ενδεχομένως αναγκάζεται να αντιδράσει [40]. Δεν τίθεται περιορισμός αν η αλλαγή λωρίδας θα είναι προς τα δεξιά ή προς τα αριστερά. Στην καθομιλουμένη αγγλική το συμβάν αυτό ονομάζεται και "cut-off", αλλά μόνο η πρώτη ονομασία χρησιμοποιείται τυπικά σε όλες τις σχετικές δημοσιεύσεις. Ο EuroNCAP θεωρεί ότι αυτό το κυκλοφοριακό φαινόμενο είναι σύνηθες και ο εν εγρηγόρσει οδηγός μπορεί να το προβλέψει και να αντιδράσει εγκαίρως ελαττώνοντας ταχύτητα. Σύμφωνα με το πρότυπο του NHTSA για την δοκιμασία αυτοκινήτων με ADS [23], η αντίληψη επικείμενης αλλαγής λωρίδας από άλλο όχημα και η αντίδραση σε αυτήν είναι απαραίτητη ικανότητα του αυτόνομου οχήματος σε κάθε τύπο δρόμου (αυτοκινητόδρομο, επαρχιακή οδό, αστική οδό).

Ο σχεδιασμός του λογικού σεναρίου από την αφηρημένη περιγραφή στηρίχθηκε στις προδιαγραφές του προτύπου ISO 34502 [20] και του πρωτοκόλλου Assisted Driving Test and Assessment του EuroNCAP [16]. Το πρότυπο ISO θεωρεί το σενάριο ολοκληρωμένο όταν η πλευρική ταχύτητα του οχήματος που κάνει τον ελιγμό μηδενιστεί.



Εικόνα 27: Σχηματική αναπαράσταση του cut-in [20].

Τα εύλογα εύρη τιμών για τα φυσικά μεγέθη της προσομοίωσης καθορίζονται εκ πείρας (το όριο ταχύτητας στον αυτοκινητόδρομο είναι μεταξύ 100 και 130km/h) αλλά και σύμφωνα με

υποδείξεις των προτύπων. Επίσης, υπάρχουν αρκετά σύνολα δεδομένων που συλλέγουν σχετικές μετρήσεις. Χαρακτηριστικό παράδειγμα είναι το "PREVENTION Dataset" [41], το οποίο έχει δημιουργηθεί οδηγώντας αυτοκίνητο εξοπλισμένο με αισθητήρες (lidar, radar, camera) σε πραγματικές οδούς υπό συνήθεις κυκλοφοριακές συνθήκες, για συνολικά 540 χιλιόμετρα. Περιέχει την πρωτογενή πληροφορία των αισθητήρων αλλά και αποτελέσματα επεξεργασίας αυτής, όπως ανίχνευση, μέσω επεξεργασίας εικόνας, αλλαγών λωρίδας, cut-ins, και άλλων φαινομένων.

Σκοπός

Ο σκοπός του σεναρίου με cut-in είναι να μελετηθούν οι διάφορες υποπεριπτώσεις του φαινομένου αυτού με δύο εμπλεκόμενα αυτοκίνητα, όπως απεικονίζεται στο πρότυπο ISO. Σε κάθε παραλλαγή αναμένεται να διαφοροποιούνται οι αρχικές θέσεις των οχημάτων, οι τελικές ταχύτητες αυτών, η διανυόμενη απόσταση κατά την αλλαγή λωρίδας, και ενδεχομένως το πλήθος λωρίδων προς αλλαγή.

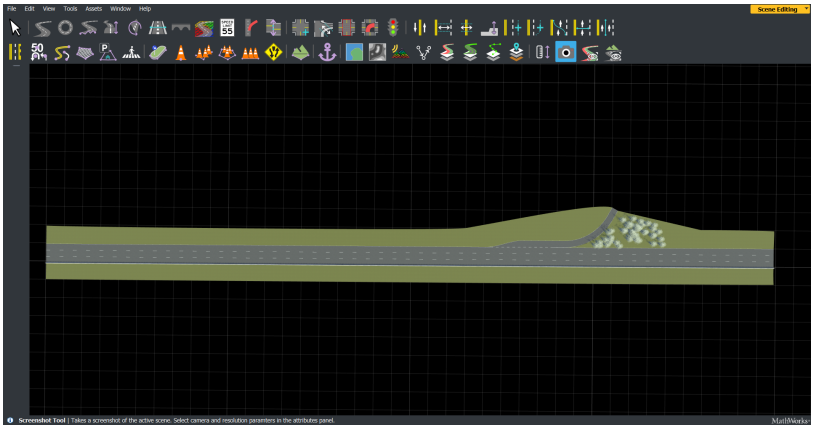
Διαφορετικά είδη

Στο πλαίσιο της εργασίας εξετάστηκε ακόμη ένα παρόμοιο σενάριο, που αφορά την είσοδο ("merging") στον αυτοκινητόδρομο από την λωρίδα εισόδου. Παρ' ότι τυπικά υπάγεται σε διακριτό είδος, πρακτικά μπορεί να αντιμετωπιστεί ως ειδική περίπτωση του cut-in, όπου το δεύτερο όχημα εισέρχεται στον αυτοκινητόδρομο από την λωρίδα εισόδου ενώ το πρώτο κινείται πίσω στην δεξιά λωρίδα. Σε αυτήν, οι ταχύτητες είναι αρκετά χαμηλότερες: για το εισερχόμενο όχημα ("on-ramp vehicle") θεωρούνται τιμές μεταξύ 40 και 50km/h.

4.1.2 Κατασκευή στο RoadRunner

Χάρτης

Ο χάρτης για το σενάριο αυτό κατασκευάστηκε με πρόθεση να επαρκεί για πάσα υποπερίπτωση αλλά ταυτοχρόνως να είναι όσο το δυνατόν μικρότερος σε μέγεθος. Βάσει των προδιαγραφών του σεναρίου, αποφασίστηκε το εικονικό περιβάλλον να είναι ένα ευθύγραμμο τμήμα αυτοκινητοδρόμου με τρεις λωρίδες, και μία λωρίδα εισόδου. Το πλήθος κυρίων λωρίδων αρμόζει σε συνήθη αυτοκινητόδρομο μικρού μεγέθους και είναι το ελάχιστο που επιτρέπει διάκριση αριστερής - μεσαίας - δεξιάς. Το μήκος του τμήματος είναι 500 μέτρα, που επαρκεί για την πραγματοποίηση του ελιγμού ακόμη και σε υψηλές ταχύτητες: ένα αυτοκίνητο κινούμενο με 120km/h διανύει 500m σε 15s, χρόνο αρκετό για την πραγματοποίηση του ελιγμού. Η απόφαση ο χάρτης να είναι κατά το δυνατόν μικρότερος έγκειται στην ανάγκη ταχείας μαζικής παραγωγής σεναρίων με αυτόν, αλλά κυρίως στην ταχύτητα φόρτωσής του από τον προσομοιωτή CARLA. Και οι δύο διαδικασίες αυτές δυσχεραίνονται από μεγαλύτερα αρχεία χάρτη.



Εικόνα 28: Κάτοψη του χάρτη στον editor του RoadRunner.

Η κατασκευή του αρχικώς στο RoadRunner ήταν απλή, καθώς αρκεί η επιλογή τύπου αυτοκινητοδρόμου από το εργαλείο χάραξης δρόμων, στην συνέχεια επιλογή σημείου έναρξης και πέρατος του δρόμου στον χώρο (κατά συνέπεια καθορισμός μήκους) και τέλος η προσθήκη της λωρίδας εισόδου με το εργαλείο slip lane και κατάλληλη προσαρμογή της διαγράμμισης.

Σενάριο

Τα δυναμικά στοιχεία του σεναρίου είναι τα δύο συμμετέχοντα αυτοκίνητα. Το ένα αυτοκίνητο έχει τον ρόλο αυτού που πραγματοποιεί το cut-in, είτε από τα αριστερά είτε από τα δεξιά του άλλου αυτοκινήτου, είτε ακόμα και από την λωρίδα εισόδου, ενώ το δεύτερο αυτοκίνητο, που είναι το υποκείμενο του σεναρίου, το επίκεντρο δηλαδή της προσομοίωσης, οφείλει να αντιδράσει στον πιθανώς επικίνδυνο ελιγμό.

Για την κατασκευή στο RoadRunner αρχικώς τοποθετούνται στον δρόμο τα δύο οχήματα από την βιβλιοθήκη γραφικών. Ύστερα καθορίζονται από το γραφικό logic editor οι αρχικές ταχύτητες αυτών, τα δυναμικά μεγέθη κατά την αλλαγή λωρίδας, και οι τελικές ταχύτητες. Συγκεκριμένα για το όχημα που πραγματοποιεί το cut-in: Η πρώτη δράση - action phase υπαγορεύει ομαλή ευθύγραμμη κίνηση με σταθερή ταχύτητα. Η επόμενη action phase, που περιγράφει την αλλαγή λωρίδας, εκκινείται μόλις εκπληρωθεί η συνθήκη τερματισμού της πρώτης. Ως συνθήκη τερματισμού χρησιμοποιείται ελάχιστη απόσταση από το πίσω αυτοκίνητο ώστε ο ελιγμός να είναι ασφαλής. Για το cut-in προσδιορίζονται οι ακόλουθες παράμετροι (attributes):

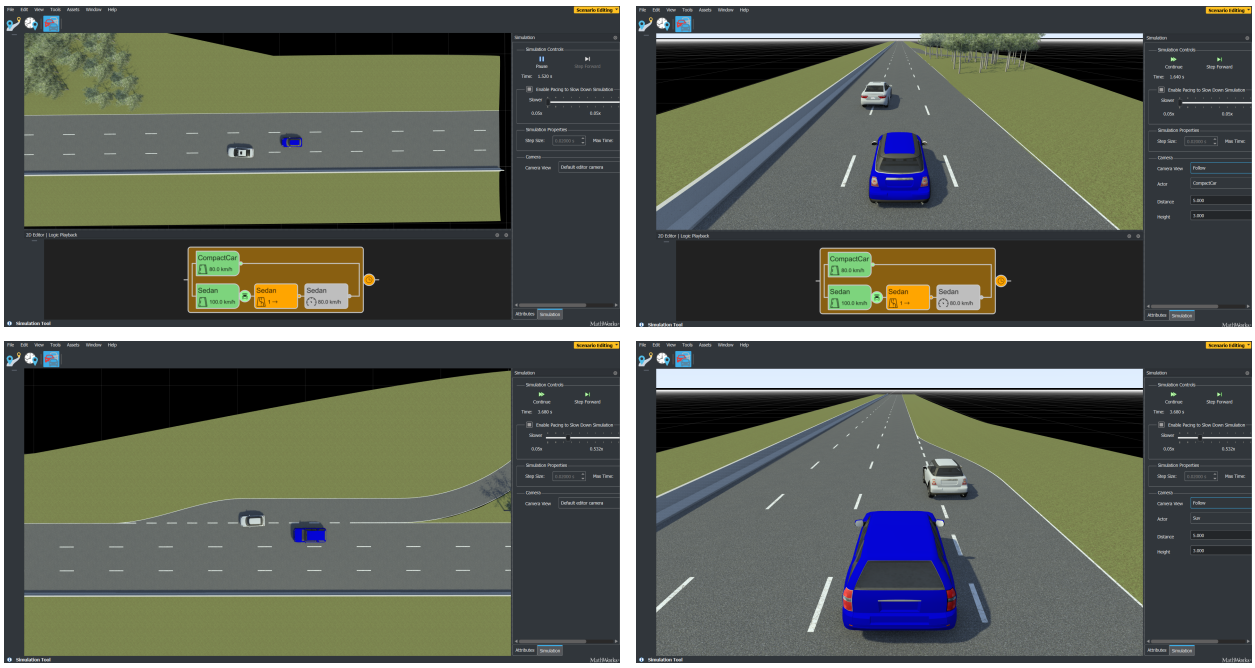
- αρχικές θέσεις και ταχύτητες των δύο αυτοκινήτων,
- κατεύθυνση αλλαγής λωρίδας (αριστερά ή δεξιά),
- πλήθος λωρίδων προς αλλαγή και
- διανυόμενη απόσταση κατά την αλλαγή.

Όπως έχει ήδη εξηγηθεί, δεν χρησιμοποιούνται οι υπόλοιπες επιλογές που παρέχει το RoadRunner για το προσδιοριζόμενο φυσικό μέγεθος κατά την αλλαγή (χρόνος, πλευρική ταχύτητα), διότι επί του παρόντος μόνο η απόσταση υποστηρίζεται από το scenario runner.

Μία περίπτωση με συγκεκριμένες τιμές

Το υποκείμενο κινείται αρχικώς στην μεσαία λωρίδα του δρόμου με 80km/h και το αυτοκίνητο που πρόκειται να κάνει το cut-in στην αριστερή με 100km/h. Όταν το δεύτερο έχει καλύψει

οριζόντια απόσταση 10m από το πρώτο, αρχίζει τον ελιγμό για να μεταβεί στην μεσαία λωρίδα. Κατά την διάρκεια αυτού διανύει περίπου 80m (αντιστοιχεί προσεγγιστικά σε διάρκεια ελιγμού τριών δευτερολέπτων). Έπειτα επιβραδύνει στα 80km/h σταδιακά σε χρονικό διάστημα 10s. Σε αυτήν την συγκεκριμένη περίπτωση το αυτοκίνητο στην μεσαία λωρίδα πιθανότατα δεν χρειάζεται να αντιδράσει αφού αντιληφθεί την πρόθεση του ταχύτερου οχήματος. Άλλες εκδοχές του ιδίου σεναρίου μπορεί να απαιτούν την απόκρισή του, αν φερ' ειπείν ο οδηγός του αυτοκινήτου που αλλάζει λωρίδα επιλέξει να μπει στην μεσαία έχοντας μικρή, μη ασφαλή απόσταση από το πίσω, ή αν επιβραδύνει απότομα.



Εικόνα 29: Δύο εκδοχές του σεναρίου σε εκτέλεση στο RoadRunner, από δύο οπτικές γωνίες.

4.1.3 Μεταφορά στον CARLA

Πρώτο βήμα είναι η κατασκευή (build) του χάρτη στην Unreal engine. Έπειτα, αφού το σενάριο έχει σχεδιαστεί στο RoadRunner και δοκιμαστεί στον ενσωματωμένο προσομοιωτή, για την εκτέλεσή του στον CARLA αρχικώς χρειάζεται εξαγωγή του αρχείου OpenSCENARIO. Ακολούθως, πρέπει να εφαρμοστούν οι αναγκαίες τροποποιήσεις που πηγάζουν από την ατελή υποστήριξη του προτύπου από τον CARLA, καθώς και ορισμένες ακόμη που εξηγούνται στην συνέχεια. Προς τον σκοπό αυτόν έχει γραφεί στο πλαίσιο της εργασίας το Python script rr_to_carla.py με το οποίο αυτοματοποιούνται όλες οι αλλαγές και παράγεται το εκτελέσιμο από τον scenario runner αρχείο OpenSCENARIO. Το script αυτό δεν είναι αρκετά γενικό για να μετατρέπει οποιοδήποτε σενάριο παραχθέν από το RoadRunner σε ερμηνεύσιμο - εκτελέσιμο από τον CARLA, καθώς οι πιθανές ασυμβατότητες είναι πολλές, αλλά λειτουργεί ειδικά για το σενάριο cut-in. Συγκεκριμένα οι αλλαγές που πραγματοποιούνται από το script είναι:

- Αλλαγή των τιμών των πεδίων filepath στις ετικέτες `LogicFile` και `SceneGraphFile` ώστε να αντιστοιχούν με το όνομα του χάρτη που έχει δοθεί κατά την εισαγωγή αυτού στην CARLA.

- Αλλαγή των πεδίων `name` των ετικετών `Vehicle` ώστε να αντιστοιχούν στα επιθυμητά μοντέλα αυτοκινήτων.
- Προσθήκη στις ίδιες ετικέτες τις υπο-ετικέτες `Property` που σηματοδοτούν τον ρόλο του κάθε οχήματος στην προσομοίωση (ego ή npc).
- Σε ορισμένες περιπτώσεις διόρθωση της ετικέτας `Orientation` κατά την αρχικοποίηση των θέσεων των οχημάτων.
- Προσθήκη κατάλληλου μπλοκ `ControllerAction` ώστε να δίνεται στον παίκτη ο έλεγχος του ego.
- Αντικατάσταση της προεπιλεγμένης συνθήκης έναρξης του σεναρίου, η οποία χρησιμοποιεί την μη υποστηριζόμενη από το scenario runner τιμή `runningState` στο πεδίο `state` του `StoryboardElementStateCondition`, με συνθήκη βάσει της απόστασης που έχει διανύσει ο hero.
- Στις συνθήκες που αφορούν μέτρηση αποστάσεως, στο πεδίο `freespace` της ετικέτας RelativeDistanceCondition επιβάλλεται η τιμή “false”.

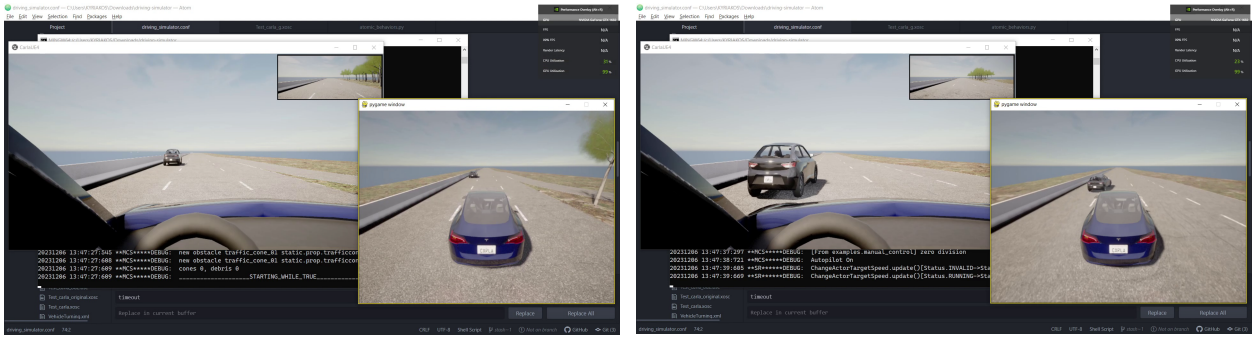
Αφού έχει προκύψει το τελικό αρχείο .xosc, για την προσομοίωση του σεναρίου στον CARLA αρκεί η μεταφορά αυτού στο προκαθορισμένο directory από όπου αντλεί τα αρχεία το scenario runner, και η συμπλήρωση του ονόματος του νέου αρχείου στην τιμή της μεταβλητής CARLA_SCENARIO στο driving_simulator.conf.

Ενώ στο RoadRunner και τα δύο οχήματα ελέγχονται από τον υπολογιστή, με αιτιοκρατικό τρόπο που καθορίζεται από τον χρήστη εξ αρχής, στον προσομοιωτή το ζήτημα είναι ακριβώς το αντίθετο, ήτοι το πίσω όχημα που καλείται να αντιδράσει στο cut-in να ελέγχεται από εξωτερικό πράκτορα (agent). Ο agent αυτός είναι είτε το υπό ανάπτυξη αυτόνομο σύστημα οδήγησης (ADS), στο πλαίσιο ανάλογης ερευνητικής εργασίας, είτε, στην περίπτωση της παρούσης εργασίας, ο άνθρωπος - χρήστης του προσομοιωτή. Στην πραγματικότητα, για τον βασικό στόχο που απασχολεί την εργασία, δηλαδή την συγγραφή σεναρίων, δεν κάνει διαφορά ποιός θα έχει τον έλεγχο του ego vehicle. Ο άνθρωπος μπορεί συμμετέχοντας στην προσομοίωση να επαληθεύσει την καλή λειτουργία του σεναρίου σύμφωνα με τις αναμενόμενες προδιαγραφές και συνεπώς την καταλληλότητα χρήσης του για την ανάπτυξη ADS, που είναι φυσικά ο τελικός σκοπός της συγγραφής σεναρίων.

Ο προσομοιωτής εκκινείται εκτελώντας το start_simulator.sh, και το scenario runner αναλαμβάνει την αρχικοποίηση των συνθηκών. Ανιχνεύει το όνομα του χάρτη από το αρχείο .xosc και στέλνει την κατάλληλη meta-command στον server για να φορτωθεί και να απεικονισθεί ο επιθυμητός χάρτης. Πρώτα ολοκληρώνεται η φόρτωση του εικονικού κόσμου και των οχημάτων στο παράθυρο του server (της Unreal) και έπειτα το ίδιο στο παράθυρο του client (PyGame). Παράλληλα στην γραμμή εντολών τυπώνονται μηνύματα (logs) για την εξέλιξη της εκτέλεσης. Μόλις ολοκληρωθεί η φόρτωση του client η προσομοίωση είναι ενεργή, το σενάριο έτοιμο προς εκτέλεση και ο χρήστης έχει τον έλεγχο του ego vehicle. Καθ' όλην την εκτέλεση του σεναρίου παράγονται σύγχρονα μηνύματα debugging από την γραμμή εντολών που ενημερώνουν για την πορεία της προσομοίωσης, ενώ όταν ολοκληρωθεί η προσομοίωση τερματίζεται ο client και παρουσιάζονται συγκεντρωτικά τα αποτελέσματα των κριτηρίων τερματισμού που ορίζονται στην συνολική `StopTrigger` στο αρχείο OpenSCENARIO.

Η ολοκλήρωση του σεναρίου επέρχεται θεωρητικά όταν περατωθεί η προσομοίωση και της τελευταίας Action που αποδίδεται σε κάποιον actor στο σενάριο. Σηματοδοτείται δε σε επίπεδο

λογισμικού από τον προσομοιωτή με την ανάθεση της τιμής `py_trees.common.Status.SUCCESS` στην μεταβλητή που αναπαριστά ανά πάσα στιγμή την κατάσταση (status) της εκτέλεσης. Στην πράξη αυτό συνεπάγεται ότι, αν παραδείγματος χάριν η τελευταία Action που ανατίθεται στο σενάριο είναι η επιτάχυνση ενός αυτοκινήτου έως κάποια τελική ταχύτητα, τότε μόλις η ταχύτητα αυτή επιτευχθεί, η εκτέλεση θα διακοπεί αμέσως. Δεν γίνεται σεβαστή, δηλαδή, από την υλοποίηση του CARLA η συνθήκη τερματισμού του σεναρίου βάσει συνολικής χρονικής διάρκειας εκτέλεσης, που τίθεται μέσω RoadRunner. Για το συγκεκριμένο πρόβλημα προτιμήθηκε η αλλαγή στην διατύπωση του σεναρίου παρά στον κώδικα του προσομοιωτή: προσθέτοντας ως τελευταία Action την επιτάχυνση ενός οχήματος κατά ελάχιστη τιμή σε μεγάλο χρονικό διάστημα (5km/h σε 60s για παράδειγμα), επιτυγχάνεται κίνηση με πρακτικά σταθερή ταχύτητα για όσον χρόνο χρειάζεται. Πρόκειται για μία παράκαμψη που αντιβαίνει στους θεωρητικούς όρους λειτουργίας του OpenSCENARIO, ωστόσο ήταν απαραίτητο να γίνουν τέτοιες προσαρμογές για την συμβατότητα με τον CARLA. Άλλωστε και σενάρια δημοσιευμένα από τον ίδιο τον οργανισμό του CARLA εφαρμόζουν παρόμοιες μεθόδους. Ακόμα, έχει ήδη εξηγηθεί ότι ορισμένες κλάσεις Atomic οριζόμενες στο atomic_behaviors.py του scenario runner, όπως η ChangeActorLateralMotion που αναλαμβάνει την εκτέλεση μίας LateralAction (τέτοια είναι η αλλαγή λωρίδας) επιστρέφουν με την ολοκλήρωσή των την τιμή `SUCCESS` επηρεάζοντας ωστόσο την γενική κατάσταση του σεναρίου και όχι μόνο της επιμέρους δράσης. Για την αντιμετώπιση αυτού του θέματος έγινε κατάλληλη τροποποίηση της συνάρτησης `update` της εν λόγω κλάσης.



Εικόνα 30: Στιγμιότυπα από την προσομοίωση.

4.1.4 Δημιουργία παραλλαγών

Η ανάγκη μελέτης όλων των διαφορετικών περιστάσεων που μπορούν να προκύψουν από διαφορές στις παραμέτρους, είναι το κίνητρο παραγωγής πολλαπλών παραλλαγών του ιδίου αυτού σεναρίου. Μέσα από το RoadRunner, κάθε μέγεθος του σεναρίου προς πειραματισμό συνδέεται με μία μεταβλητή. Με την χρήση του gRPC API του RoadRunner, ένα Python script μπορεί να προσπελάσει και να επεξεργαστεί τις τιμές των μεταβλητών αυτών, και έπειτα να κινητοποιήσει αυτόματα την διαδικασία εξαγωγής του σεναρίου σε OpenSCENARIO. Οι μέθοδοι που πραγματοποιούν αυτές τις λειτουργίες προέρχονται από ειδική βιβλιοθήκη της mathworks, η οποία παρέχεται μαζί με την εγκατάσταση του RoadRunner αλλά απαιτεί μεταγλώττιση για να χρησιμοποιηθεί από την Python [19][34].

Για την συστηματοποίηση της διαδικασίας παραγωγής παραλλαγών του σεναρίου αναπτύχθηκε jupyter notebook που εκτελεί με την σειρά τα απαραίτητα βήματα. Η επιλογή του jupyter έγκειται στην ευκολία χρήσης και διαδραστικότητα που παρέχει, δίνοντας ανατροφοδότηση μετά από κάθε κελλί κώδικα και επιτρέποντας την εκτέλεση των επιμέρους βημάτων χωριστά. Στο notebook Generate_variations_and_export.ipynb αρχικώς αξιοποιείται η γραμμή εντολών των Windows μέσω της βιβλιοθήκης os της Python για την έμμεση εκτέλεση εντολών που εκκινούν τον RoadRunner - server του μοντέλου gRPC και φορτώνουν το επιθυμητό σενάριο σε αυτόν. Ο χρήστης καθορίζει τα ονόματα των μεταβλητών του σεναρίου με των οποίων τις τιμές επιθυμεί να πειραματιστεί, καθώς και τα (πεπερασμένα) σύνολα τιμών που θα λάβουν αυτές. Κατόπιν αναλαμβάνει το μέρος του script που έχει ουσιαστικά τον ρόλο του client στο μοντέλο gRPC. Για κάθε δυνατό συνδυασμό τιμών όλων των μεταβλητών, το πρόγραμμα με τις κατάλληλες συναρτήσεις πρώτα αποστέλλει αιτήματα αλλαγής των τιμών στις μεταβλητές του RoadRunner και ύστερα αποστέλλει αίτημα εξαγωγής του σεναρίου με τις εκάστοτε τιμές σε OpenSCENARIO. Αφού όλες οι δυνατές παραλλαγές έχουν παραχθεί, με την χρήση του rr_to_carla.py το πρόγραμμα τελικώς δημιουργεί τα εκτελέσιμα από τον CARLA αρχεία .xosc και τα αποθηκεύει στην κατάλληλη τοποθεσία.

Η συνάρτηση `set_variable` επικοινωνεί μέσω του API με τον server του RoadRunner και αλλάζει τιμή παραμέτρου

[6]:
```
def set_variable(variable_name, value):
    ## Connect to RoadRunner API server
    with grpc.insecure_channel('localhost:' + str(PORT)) as channel:
        api = roadrunner_service_pb2_grpc.RoadRunnerServiceStub(channel)
        setVariableRequest = roadrunner_service_messages_pb2.SetVariableRequest()
        setVariableRequest.name = variable_name
        setVariableRequest.value = value
        api.SetScenarioVariable(setVariableRequest)
```

Κατασκευάζω τον κατάλογο των παραμέτρων του σεναρίου με τις οποίες θέλω να πειραματιστώ (να κατασκευάσω παραλλαγές) και τα επιθυμητά διαστήματα τιμών αυτών

[7]:
```
RoadRunnerVariables = dict()
RoadRunnerVariables['SedanTargetSpeed1'] = range(10, 30, 10)
RoadRunnerVariables['SedanForwardOffset'] = range(15, 30, 10)
RoadRunnerVariables['ChangeLaneOverDistance'] = range(30, 60, 10)
```

Εικόνα 31: Απόσπασμα από το notebook όπου ορίζεται η μέθοδος αλλαγής τιμών variables του RoadRunner.

Για κάθε δυνατό συνδυασμό παραμέτρων, αλλαγή των αντίστοιχων μεταβλητών στο scenario RoadRunner και εξαγωγή σε μορφή OpenSCENARIO 1.x

[12]:
```
for permutation in itertools.product(*RoadRunnerVariables.values()):
    filename = '_'.join(['_'.join([n, str(v)]) for n, v in zip(RoadRunnerVariables.keys(), permutation)])
    osc_files.append(filename)
    print(filename)
    for param, val in zip(RoadRunnerVariables.keys(), permutation):
        set_variable(param, str(val))
    os.system('mkdir ' + EXPORT_PATH.replace('/', '\\') + '\\' + filename)
    os.system(f'CmdRoadRunnerAPI "Export(file_path=\'{EXPORT_PATH}/{filename}/{filename}\' format_name=\'OpenSCENARIO\')" --serverAddress=localhost:{PORT
```

```
SedanTargetSpeed1_10_SedanForwardOffset_15_ChangeLaneOverDistance_30
SedanTargetSpeed1_10_SedanForwardOffset_15_ChangeLaneOverDistance_40
SedanTargetSpeed1_10_SedanForwardOffset_15_ChangeLaneOverDistance_50
SedanTargetSpeed1_10_SedanForwardOffset_25_ChangeLaneOverDistance_30
SedanTargetSpeed1_10_SedanForwardOffset_25_ChangeLaneOverDistance_40
SedanTargetSpeed1_10_SedanForwardOffset_25_ChangeLaneOverDistance_50
SedanTargetSpeed1_20_SedanForwardOffset_15_ChangeLaneOverDistance_30
SedanTargetSpeed1_20_SedanForwardOffset_15_ChangeLaneOverDistance_40
SedanTargetSpeed1_20_SedanForwardOffset_15_ChangeLaneOverDistance_50
SedanTargetSpeed1_20_SedanForwardOffset_25_ChangeLaneOverDistance_30
SedanTargetSpeed1_20_SedanForwardOffset_25_ChangeLaneOverDistance_40
SedanTargetSpeed1_20_SedanForwardOffset_25_ChangeLaneOverDistance_50
```

Εικόνα 32: Ο κώδικας παραγωγής των παραλλαγών.

Συνοψίζοντας, η αυτοματοποίηση της συγγραφής παραλλαγών ενός σεναρίου αυτοματοποιείται, απαιτώντας από τον χρήστη μόνο τον προσδιορισμό των τιμών των παραμέτρων προς επεξεργασία, εφ' όσον έχουν προηγηθεί τα εξής βήματα:

- Σχεδιασμός του χάρτη στο RoadRunner.
- Εξαγωγή και build του ιδίου χάρτη στην Unreal για τον CARLA.
- Σχεδιασμός του σεναρίου στο RoadRunner και δοκιμή αυτού.
- Ανάθεση μεταβλητών στο RoadRunner στις κρίσιμες παραμέτρους - φυσικά μεγέθη του σεναρίου.

4.1.5 Συμπέρασμα

Πολλά σενάρια των πρωτοκόλλων ασφαλείας περιγράφουν σύντομες αλληλεπιδράσεις μεταξύ ολιγάριθμων οχημάτων, καθοριζόμενες από διαχειρίσιμο πλήθος φυσικών παραμέτρων. Φαίνεται πως το RoadRunner είναι απολύτως επαρκές για τον σχεδιασμό τέτοιων σεναρίων και επιτρέπει τον πειραματισμό με τις παραμέτρους παρέχοντας εύχρηστη προγραμματιστική διεπαφή για την συγγραφή διαφόρων εκδοχών. Ο προσομοιωτής CARLA δεν υποστηρίζει όλες τις επιλογές για περιγραφή των κινητικών και δυναμικών μεγεθών υπαγορευόμενες από το πρότυπο OpenSCENARIO, ωστόσο υποστηρίζει αρκετές ώστε να μπορεί να εκτελεί τέτοια σενάρια. Αναμένεται να βελτιωθεί ως προς αυτό αφού πρόκειται για αναπτυσσόμενο έργο. Επιπλέον, διαπιστώνεται ότι RoadRunner και CARLA διαφέρουν σε ελάχιστα σημεία ως προς την σύνταξη OpenSCENARIO που υποθέτουν, αλλά οι διαφορές αυτές είναι επιλύσιμες προγραμματιστικά.

4.2 Αυτοκινητοπομπή σε κυκλικό κόμβο

4.2.1 Περιγραφή

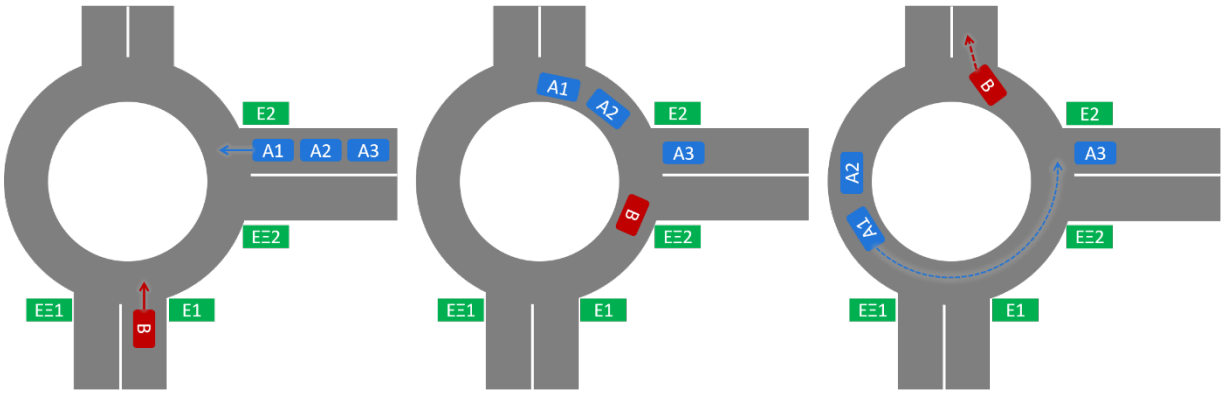
Σκοπός είναι η εξέταση του RoadRunner ως προς την ικανότητα παραγωγής σεναρίων στα οποία οι κινήσεις των οχημάτων προσδιορίζονται χαράσσοντας επακριβώς τροχιές, όχι προγραμματίζοντας εντολές καθορισμού ταχύτητας και λοιπών κινητικών μεγεθών. Παράλληλα, σε αντιδιαστολή με το προηγούμενο σενάριο, το παρόν διεξάγεται σε αστικό περιβάλλον και συμμετέχουν σε αυτό περισσότεροι πράκτορες.

Το σενάριο

Το επιλεγμένο σενάριο έχει σχεδιαστεί σε λογικό επίπεδο και ερευνάται από το τεχνικό πρόγραμμα (project) EVENTS. Το πρόγραμμα αυτό είναι συνεργασία μεταξύ οργανισμών ευρωπαϊκών χωρών (συμπεριλαμβανομένου και του ΕΠΙΣΕΥ) και χρηματοδοτούμενο από την Ευρωπαϊκή Ένωση μελετά περιπτώσεις διατάραξης της προγραμματισμένης λειτουργίας αυτόνομων οχημάτων. Ενδιαφέρεται ειδικά για την εξέταση της αντιληπτικής ικανότητας (perception) και της λήψης αποφάσεων (“decision making”) του ADS υπό δυσχερείς συνθήκες όπως ανεπάρκεια φωτός, βροχή και οπτικά εμπόδια [42].

Εν προκειμένω το σενάριο λαμβάνει χώρα σε έναν κυκλικό κόμβο, και συμμετέχουν σε αυτό τέσσερα αυτοκίνητα. Τρία εξ αυτών κινούνται μαζί σε σχηματισμό αυτοκινητοπομπής (convoy) και οφείλουν να τηρούν τον σχηματισμό αυτόν. Το τέταρτο αναγκάζει τον σχηματισμό σε ρήξη και συνεπώς πρέπει να γίνουν οι κατάλληλες κινήσεις για να επανέλθουν μετά τον κόμβο. Συγκεκριμένα η σειρά των γεγονότων περιγράφεται ως εξής:

1. Τα A1, A2, A3 εισέρχονται σε σχηματισμό στον κυκλικό κόμβο από την είσοδο E2 με πρόθεση να εξέλθουν στην ΕΞ1.
2. Το B κινείται προς την είσοδο E1. Η έξοδος που σκοπεύει να πάρει δεν είναι γνωστή.
3. Τα A1 και A2 εισέρχονται με ασφάλεια στον κόμβο.
4. Το B εισέρχεται στον κόμβο.
5. Το A3 εντοπίζει το B, κρίνει ότι δεν προλαβαίνει να εισέλθει και του παραχωρεί προτεραιότητα, ενώ τα A1 και A2 παραμένουν σε σχηματισμό.
6. Τα A1 και A2 προσπερνούν την έξοδο και παραμένουν εντός του κόμβου έως ότου το A3 επανασυνδεθεί στην αυτοκινητοπομπή.



Εικόνα 33: Σχηματική αναπαράσταση του σεναρίου convoy.

Το θέμα της προτεραιότητας στον κυκλικό κόμβο

Ο κυκλικός κόμβος (roundabout) είναι στοιχείο του οδικού δικτύου όπου δύο ή περισσότεροι δρόμοι μονής ή διπλής κατευθύνσεως συγκλίνουν σε διασταύρωση κυκλικού σχήματος γύρω από κεντρική νησίδα, στην οποία επιτρέπεται η κυκλοφοριακή ροή προς μία μόνο κατεύθυνση. Στις περισσότερες χώρες του κόσμου, το εισερχόμενο όχημα υποχρεούται να παραχωρήσει προτεραιότητα στο οχήματα που κινούνται ήδη μέσα στον κυκλικό κόμβο. Έχει παρατηρηθεί ότι αυτός ο κανόνας ευνοεί την κυκλοφοριακή ροή. [43]

Αντιθέτως, στην Ελλάδα ο Κώδικας Οδικής Κυκλοφορίας [44] δεν αναφέρεται συγκεκριμένα στην κυκλική διασταύρωση, και συνεπώς για αυτήν ισχύει ο γενικός κανόνας για τους ισόπεδους κόμβους, από το Άρθρο 26, παράγραφος 4: «Στους κόμβους η προτεραιότητα ορίζεται με κατάλληλη σήμανση» και παράγραφος 5: «Στους κόμβους χωρίς τέτοια σήμανση η προτεραιότητα ανήκει σε αυτόν που έρχεται από τα δεξιά». Επειδή στις χώρες που η πλευρά οδήγησης είναι η δεξιά, η φορά τον κυκλικών κόμβων είναι αντίστροφη ωρολογιακή, αποτέλεσμα του κανόνα είναι η προτεραιότητα να δίνεται στους εισερχομένους στον κόμβο.

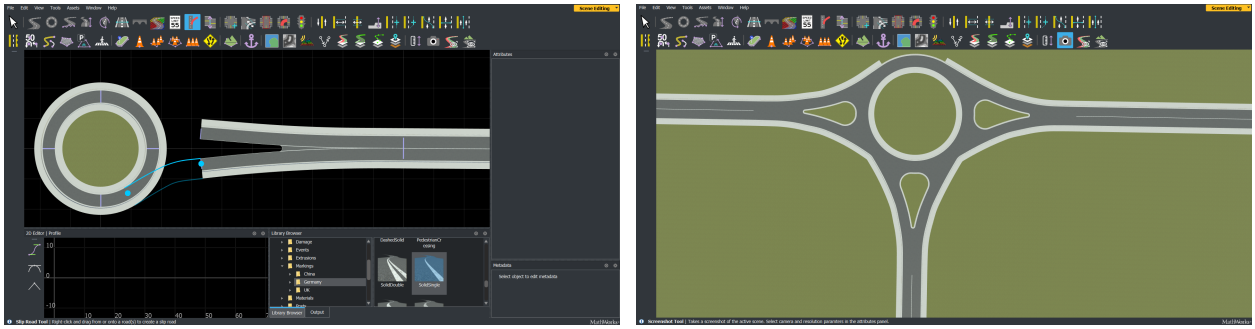
Για αυτό το μέρος της εργασίας, βάσει των προδιαγραφών του σεναρίου, έχει υποτεθεί ότι ισχύει αντί του ΚΟΚ η προτεραιότητα στο όχημα εντός του κόμβου.

4.2.2 Κατασκευή στο RoadRunner

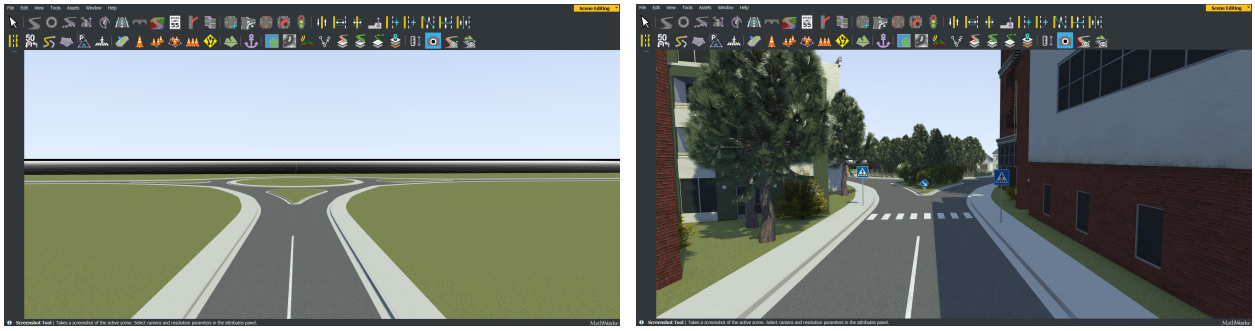
Σχεδιασμός χάρτη

Στην περιγραφή του EVENTS δεν καθορίζεται σαφώς η μορφή του χάρτη, οπότε επελέγη μία σχετικά απλή διάταξη κυκλικού κόμβου. Στον κόμβο συμβάλλουν σε σχήμα «Τ» τρεις δρόμοι διπλής κατεύθυνσης με μία λωρίδα ανά κατεύθυνση, ενώ ο ίδιος ο κυκλικός δρόμος διαθέτει επίσης μία λωρίδα. Η εσωτερική διάμετρος του κυκλικού δρόμου είναι περίπου 20m, μέγεθος σύνηθες για τέτοιες διασταυρώσεις εντός αστικού ιστού.

Με βάση το ίδιο αυτό οδικό δίκτυο κατασκευάστηκαν δύο χάρτες. Ο πρώτος περιέχει μόνο τους δρόμους και τις πρακτικά ισόπεδες με την άσφαλτο νησίδες. Στον δεύτερο προστέθηκαν πολλά επιπλέον στοιχεία περιβάλλοντος προκειμένου να προσομοιωθούν πραγματικές συνθήκες αστικού χώρου και κυρίως να υπάρχουν αρκετά οπτικά εμπόδια για την δοκιμασία της αντίληψης του αυτόνομου συστήματος. Συγκεκριμένα τοποθετήθηκαν κτήρια στα οικοδομικά τετράγωνα γύρω από τον κόμβο, πυκνή βλάστηση στην κυκλική νησίδα και πινακίδες κυκλοφορίας. Όλα αυτά τα επιπλέον τρισδιάστατα γραφικά μοντέλα του RoadRunner πέρασαν επιτυχώς και κατασκευάστηκαν στην Unreal κατά την μεταφορά του χάρτη, με εξαίρεση τις επιφάνειες των πινακίδων. Επειδή οι δύο εκδοχές του χάρτη εδράζονται στο ίδιο ακριβώς οδικό δίκτυο και ουσιαστικά διαφέρουν μόνο στα πρόσθετα στοιχεία, ένα σενάριο που έχει συγγραφεί για τον πρώτο μπορεί να εκτελεστεί αυτούσιο και στον δεύτερο και αντιστρόφως. Αυτό επειδή, όπως είναι κατασκευασμένοι οι χάρτες, τόσο αρχικά στο RoadRunner όσο και στην συνέχεια στην Unreal, κάθε σημείο του χώρου στον έναν αντιστοιχίζεται βάσει συντεταγμένων στο ίδιο ακριβώς σημείο στον άλλον. Συνεπώς οι θέσεις και οι κινήσεις των οχημάτων στον δρόμο που ορίζονται στο σενάριο θα αποτυπώνονται ίδιες και στους δύο χάρτες.



Εικόνα 34: Δύο κατόψεις. Αριστερά από την κατασκευή του χάρτη, δεξιά η τελική μορφή.



Εικόνα 35: Προοπτική των δύο εκδοχών στο RoadRunner από το ίδιο σημείο θέασης.

Συγγραφή σεναρίου

Οποιοδήποτε από τα τρία αυτοκίνητα της αυτοκινητοπομπής θα μπορούσε να είναι το όχημα υπό εξέταση. Σε μία πλήρη προσομοίωση και τα τρία θα ήταν ego vehicles ελεγχόμενα από αυτόνομο σύστημα και θα δοκιμάζονταν ταυτοχρόνως. Για την εργασία επελέγη ως επίκεντρο το τρίτο αυτοκίνητο. Αυτό σημαίνει ότι κατά την προσομοίωση ο χρήστης πλοηγεί το όχημα αυτό ενώ η συμπεριφορά των άλλων τριών είναι προκαθορισμένη. Από αυτήν την οπτική γωνία, όταν στην προσομοίωση το όχημα το χειρίζεται το ADS, ελέγχεται η ικανότητα αυτού αρχικώς να ανιχνεύσει το κινούμενο όχημα εντός του κόμβου, να λάβει την απόφαση να παραχωρήσει προτεραιότητα, στην συνέχεια να περιμένει να εντοπίσει την επιστροφή των άλλων δύο μελών της συνοδείας και να εισέλθει στον κόμβο την κατάλληλη στιγμή.

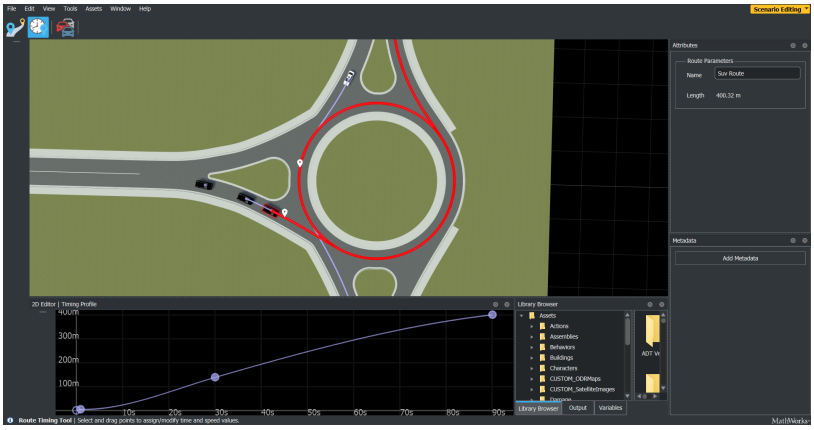
Για την υλοποίηση του σεναρίου στο RoadRunner απαιτείται η χρήση δράσεων path following, οι οποίες σε OpenSCENARIO μεταφράζονται σε FollowTrajectoryAction. Η ακριβής χάραξη της τροχιάς του οχήματος είναι απαραίτητη διότι δεν υπάρχει ούτε στο RoadRunner αλλά ούτε και στο πρότυπο OpenSCENARIO ειδική Action που να επιβάλλει την στροφή από έναν δρόμο σε έναν άλλον. Συνεπώς, κανένας τύπος Action εκτός από τον FollowTrajectory δεν μπορεί να επιβάλλει συμπεριφορά οχήματος διαφορετική από την προεπιλεγμένη, τουτέστιν την σταθερή παραμονή στον ίδιο δρόμο (βλ. Κεφάλαιο 2). Φυσικά με αυτόν τον κανόνα δεν γίνεται να διεξαχθεί το σενάριο, καθώς αναμένεται από τα οχήματα αφού εισέλθουν στον κυκλικό κόμβο να πάρουν κάποια έξοδο, οπότε να αλλάξουν οδό.

Η χρήση path following επιβάλλει έναν περιορισμό: επειδή κάθε τροχιά έχει σημείο έναρξης και λήξης με ακριβείς συντεταγμένες, δεν επιτρέπεται να προγραμματιστεί άλλη action phase (όπως επιτάχυνση) στο διάγραμμα ροής του αυτοκινήτου πριν από την FollowTrajectory. Επιπλέον, μέσα σε μία δράση path following δεν εντάσσονται λογικές συνθήκες για την

προσομοίωση λήψης αποφάσεων από τα οχήματα, όπως γίνεται μεταξύ action phases στο σενάριο cut-in. Επειδή όμως, όπως αναλύεται στην συνέχεια, η ίδια η χάραξη τροχιάς στο RoadRunner παρέχει υψηλή προσαρμοστικότητα, είναι εφικτό με αυτήν την μία Action να προσομοιωθεί σχεδόν οποιαδήποτε σειρά κινήσεων.

Ο σχεδιασμός μίας τροχιάς κίνησης για ένα αυτοκίνητο στο RoadRunner πραγματοποιείται τοποθετώντας σημεία αναφοράς (waypoints) επάνω στον χάρτη. Αυτά συνδέονται με γραμμή με την σειρά που τοποθετούνται και συντίθεται η συνεχής τροχιά. Όταν ένα waypoint τοποθετείται επάνω στο οδικό δίκτυο, η γραμμή που το συνδέει με το προηγούμενο ακολουθεί την λογική γραμμή του δρόμου, περιγράφοντας κανονική οδήγηση από το προηγούμενο σημείο στο επόμενο. Το RoadRunner επιτρέπει την επεξεργασία της γραμμής σε οποιοδήποτε σχήμα, και επίσης την τοποθέτηση waypoints εκτός δρόμου, αλλά αυτές οι δυνατότητες δεν εξυπηρετούν τον σκοπό ούτε απασχολούν την παρούσα εργασία.

Στον κυκλικό κόμβο, τοποθετώντας τα waypoints με την σειρά ώστε η τροχιά να κάνει δύο φορές τον κύκλο επιτυγχάνεται η προσομοίωση της επιθυμητής συμπεριφοράς των προπορευόμενων οχημάτων του convoy. Για κάθε waypoint, το RoadRunner διαθέτει μέτρηση της απόστασης από την αφετηρία και επιτρέπει στον χρήστη τον προσδιορισμό της χρονικής στιγμής και της ταχύτητας άφιξης σε αυτό. Λαμβάνοντας αυτά τα δεδομένα για κάθε waypoint, το RoadRunner αναλαμβάνει αυτομάτως την καμπύλη θέσης-χρόνου του αυτοκινήτου για τα διαστήματα μεταξύ των σημείων. Επομένως ο χρήστης μπορεί να καθορίσει την συνολική διάρκεια της κίνησης και την ταχύτητα του οχήματος σε οποιοδήποτε σημείο της τροχιάς τον ενδιαφέρει.



Εικόνα 36: Σχεδιασμός του σεναρίου με χάραξη τροχιάς και καμπύλη χρόνου-θέσης για τα waypoints.

4.2.3 Μεταφορά στον CARLA

Εξαγωγή σε OpenSCENARIO

Όταν το σενάριο από RoadRunner μετατρέπεται σε OpenSCENARIO, οι πληροφορίες για τις θέσεις των waypoints και οι ταχύτητες και χρονοσφραγίδες που έχει ορίσει ο χρήστης δεν κωδικοποιούνται, αλλά το RoadRunner τις ενσωματώνει μέσα από την ακολουθία κορυφών (ετικετών `<Vertex>`) που απαρτίζουν την γραμμή της κάθε τροχιάς. Μία Vertex αποτελείται από μία τιμή χρόνου και τις συντεταγμένες ενός σημείο στον χώρο. Το RoadRunner αναθέτει

στις ετικέτες Vertex τις κατάλληλες τιμές στα πεδία αυτά, ώστε ένα σώμα, ακολουθώντας τα αναπαριστώμενα σημεία στον χώρο με τον ρυθμό που υπαγορεύουν οι χρονοσφραγίδες, να εκτελεί ακριβώς την κίνηση που έχει περιγράψει ο χρήστης.

Επειδή η `FollowTrajectoryAction` είναι η πρώτη (και στο συγκεκριμένο σενάριο μοναδική) Action κάθε οχήματος, στο αρχείο .xosc το RoadRunner την τοποθετεί στην περιοχή `<Init>` του `<Storyboard>` αντί της περιοχής `<Story>`. Αυτή η σύνταξη δεν αντιβαίνει στο πρότυπο OpenSCENARIO, ωστόσο έχει ως αποτέλεσμα όταν το σενάριο εκτελείται από τον CARLA, η κίνηση αυτή να πραγματοποιείται αμέσως μόλις φορτώσει ο server και όχι αφού έχει φορτώσει και ο client, γεγονός προφανώς ανεπιθύμητο. Το πρόβλημα αυτό λύνεται δημιουργώντας στην περιοχή `<Story>` μία `<ManeuverGroup>` για κάθε actor, εντός της οποίας μεταφέρεται η αντίστοιχη `<FollowTrajectoryAction>` και στην θέση αυτής στην Init καθορίζονται απλώς οι αρχικές θέσεις των οχημάτων. Αυτή είναι η μοναδική επιπλέον αναγκαία αλλαγή στο αρχείο .xosc πέραν όσων περιγράφονται στην παράγραφο 4.1.

Υλοποίηση από scenario runner

Κατ' αρχάς, η πρώτη έκδοση του λογισμικού scenario runner που υποστηρίζει την κίνηση βάσει τροχιάς είναι η 0.9.15, ενώ ο κώδικας του driving-simulator του εργαστηρίου χρησιμοποιεί προηγούμενη έκδοση. Για να εκτελεσθεί το σενάριο χρειάστηκε να μεταφερθούν από την νεώτερη έκδοση μόνο οι προσαρμογές στα αρχεία openscenario_parser.py και atomic_behaviors.py

Την εκτέλεση της `FollowTrajectoryAction` στην προσομοίωση την αναλαμβάνει η κλάση `ChangeActorWaypoints` του atomic_behaviors.py. Κάθε αντικείμενο της κλάσης διατηρεί λίστα των vertices της αντίστοιχης `Trajectory` που έχει προκύψει από το parsing του αρχείου .xosc. Η συνάρτηση `update` της κλάσης σε κάθε παλμό υπολογίζει την ταχύτητα που χρειάζεται το όχημα για να φθάσει εγκαίρως στην επόμενη Vertex και εφαρμόζει την ταχύτητα αυτήν καλώντας την συνάρτηση `update_target_speed`. Συγκεκριμένα, για τον υπολογισμό της ταχύτητας χρησιμοποιείται διαφορική μέθοδος: σε κάθε παλμό που εκτελείται η `update`, λαμβάνει την θέση του οχήματος από την `CarlaDataProvider` και υπολογίζει βάσει του χρονομέτρου της προσομοίωσης `GameTime.get_time()` και των χρονοσφραγιδών των Vertex την επόμενη θέση της τροχιάς στην οποία θα πρέπει να φθάσει το όχημα (την πρώτη θέση της τροχιάς με χρονική στιγμή ύστερη της παροντικής). Έπειτα, έχοντας την τρέχουσα θέση και την επόμενη αναμενόμενη θέση, υπολογίζει τον υπολειπόμενο χρόνο για την πραγματοποίηση της μετατόπισης, ως διαφορά της χρονοσφραγίδας της τελικής Vertex από την τρέχουσα χρονική στιγμή. Τέλος, η απαραίτητη ταχύτητα προκύπτει (στην συνάρτηση `_update_speed`) ως το πηλίκο της απόστασης (σε ευθεία εφ' όσον πρόκειται για διαφορική μέθοδο και η απόσταση μεταξύ σημείων είναι ελάχιστη) της τρέχουσας θέσης από την τελική διά τον υπολειπόμενο χρόνο.

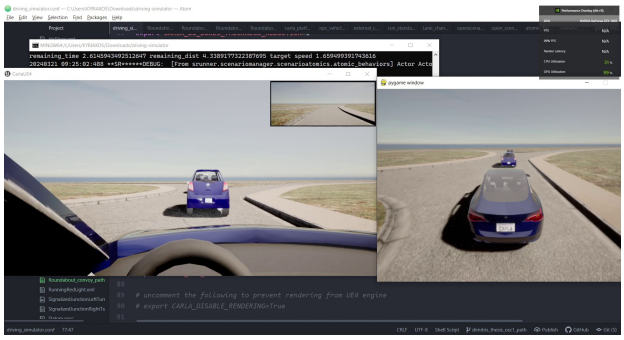
Η μέθοδος περιεγράφη αναλυτικά ώστε να αναδειχθεί το πρόβλημα που προκύπτει εξ αιτίας της: παρ' ότι θεωρητικά είναι λογική και οδηγεί στην σωστή κίνηση, στην πράξη παρατηρείται ότι τα αυτοκίνητα μένουν πίσω εν σχέσει με το χρονοπρόγραμμα της πορείας, με αποτέλεσμα ο παρονομαστής του κλάσματος να είναι πολύ μικρός και συνεπώς να προκύπτουν παράλογες στιγμιαίες ταχύτητες της τάξεως των χιλιάδων m/s. Τελικώς από την Unreal οι υπερβολικά υψηλές τιμές αυτές δεν προσομοιώνονται ακριβώς, ωστόσο παρατηρούνται απότομες

επιταχύνσεις και το δεύτερο αυτοκίνητο του convoy προσκρούει από πίσω στο προπορευόμενο. Για την διόρθωση ολόκληρης αυτής της αστοχίας, με μικρή παραλλαγή στον κώδικα τίθεται μέγιστη επιτρεπτή τιμή 6m/s = 21.6km/h για την `target_speed` στην συνάρτηση `_update_speed`. Το όριο θα μπορούσε να είναι υψηλότερο εντός του επιτρεπτού πλαισίου για αστικές οδούς· η σχετικά χαμηλή τιμή επελέγη ώστε, αφ' ενός η προσομοίωση να διεξάγεται αργά ώστε να παρατηρείται ευκολότερα η κίνηση και συμπεριφορά κάθε οχήματος και αφ' ετέρου για να δοκιμαστεί η λειτουργία της διαφορικής μεθόδου σε ακραίες τιμές. Στην προσομοίωση τα αυτοκίνητα ακολουθούν τις τροχιές κανονικά, οπότε η αλλαγή αυτήν στον κώδικα διατηρεί την καλή λειτουργία της διαφορικής μεθόδου.

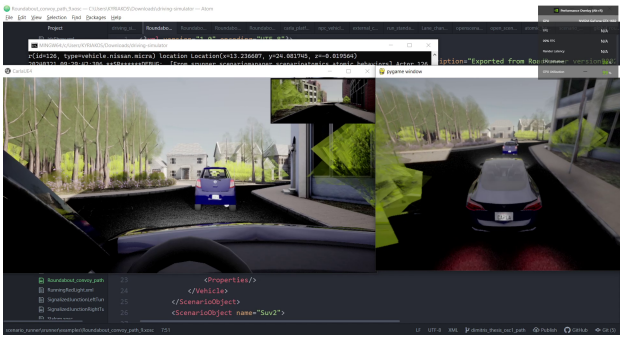
```
879            if self._times is not None:
880                current_relative_time = GameTime.get_time() - self._start_time
881                current_waypoint_idx = bisect_right(self._times, current_relative_time)
882                if current_waypoint_idx >= len(self._times):
883                    return py_trees.common.Status.SUCCESS
884                remaining_time = self._times[current_waypoint_idx] - current_relative_time
885                self._update_speed(actor, self._waypoints[current_waypoint_idx], remaining_time)
886
887            return py_trees.common.Status.RUNNING
888
889        def _update_speed(self, actor, target_waypoint, remaining_time):
890            target_location = sr_tools.openscenario_parser.OpenScenarioParser.convert_position_to_transform(
891                target_waypoint[0]).location
892            remaining_dist = calculate_distance(CarlaDataProvider.get_location(self._actor), target_location)
893            target_speed = remaining_dist / max(remaining_time, 0.001)
894            if(self._actor.id == 126):
895                logging.debug(f'[From {__name__}] Actor 126 remaining_time {remaining_time} remaining_dist {remaining_d
896            if(target_speed > 6):
897                target_speed = 6
898            actor.update_target_speed(target_speed)
```

Εικόνα 37: Το επίμαχο τμήμα κώδικα της ChangeActorWaypoint στο atomic_behaviors.py.

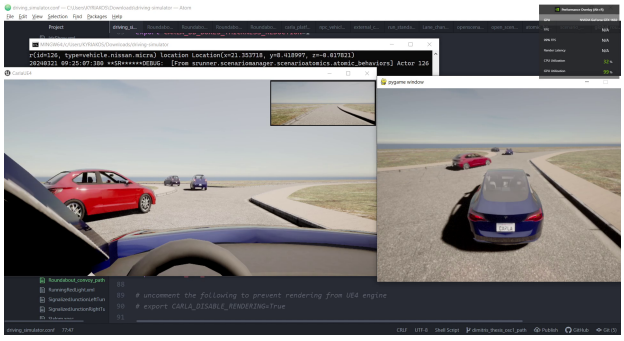
Ακόμη μία παρατήρηση για την υλοποίηση του path following είναι ότι στην πραγματικότητα δεν χρησιμοποιούνται όλα τα δεδομένα σημείων Vertex που περιέχονται στο αρχείο OpenSCENARIO, μίας και η συνάρτηση σε κάθε παλμό εξετάζει το πλησιέστερο χρονικά σημείο χωρίς να εγγυάται ότι δεν έχει προσπεράσει προηγούμενα. Ωστόσο, δεν φαίνεται πως αυτός ο σχεδιασμός επιφέρει αλλαγή στην τροχιά, μιας και αυτή αποτελείται από πολύ μεγάλο σύνολο Vertex. Ενδεικτικά, η τροχιά για το τέταρτο αυτοκίνητο του σεναρίου, που διανύει απόσταση περίπου 154 μέτρων αποτελείται από 141 σημεία.



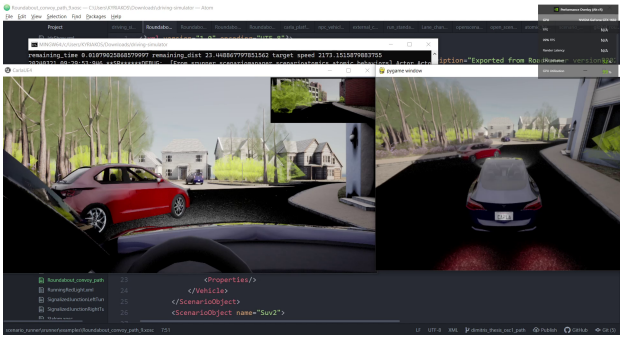
(1α)



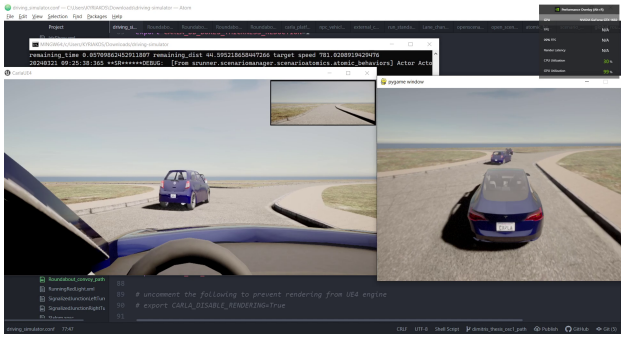
(1β)



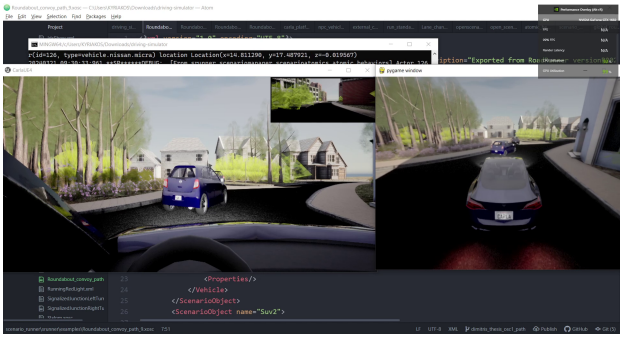
(2α)



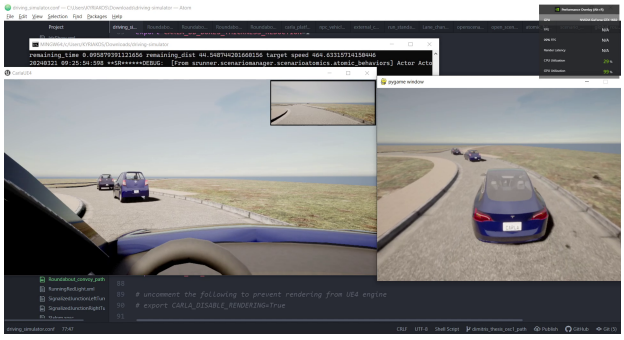
(2β)



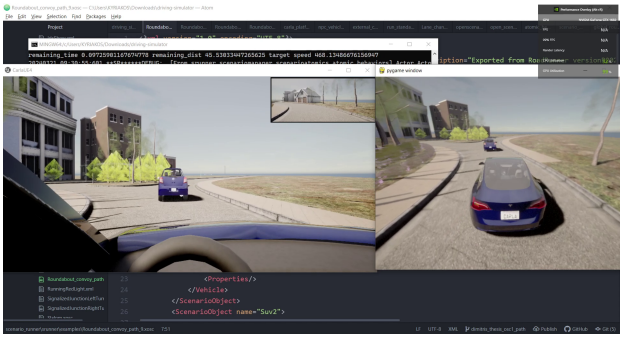
(3α)



(3β)



(4α)



(4β)

Εικόνα 38: Διαδοχικά στιγμιότυπα από το σενάριο στους δύο χάρτες. 1: είσοδος στον κόμβο, 2: παρεμβολή ξένου οχήματος, 3: τα δύο πρώτα επιστρέφουν στο τρίτο που περιμένει, 1: η πορεία συνεχίζεται.

4.2.4 Συμπέρασμα

Φαίνεται πως το RoadRunner (και κατ' επέκτασιν το πρότυπο OpenSCENARIO) δεν είναι απολύτως επαρκές για την περιγραφή τέτοιων συνθέτων σεναρίων. Ο βασικός λόγος είναι πως αναγκάζει την χρήση δράσεων path following, στις οποίες δεν εντάσσονται λογικές συνθήκες μεταξύ των σημείων αναφοράς. Επιπλέον, βάσει του προτύπου η τροχιά είναι δεσμευμένη στον χώρο, αφού τα ίδια τα σημεία αναφοράς είναι πακτωμένα σε συγκεκριμένες συντεταγμένες, που συνεπάγεται εξαιρετικά περιορισμένη δυνατότητα τροποποίησης του σεναρίου. Με αυτούς τους περιορισμούς, δεν κρίνεται χρήσιμη η αυτοματοποιημένη παραγωγή παραλλαγών ενός σεναρίου. Ωστόσο, με τις απαραίτητες προσαρμογές στον προσομοιωτή CARLA, είναι εφικτή η συγγραφή μεμονωμένου σεναρίου με περιορισμένες δυνατότητες όμως αρκετά περίπλοκου ώστε να προσφέρει κάποια αξία στην έρευνα.

Κεφάλαιο 5: Επίλογος

5.1 Πορίσματα

Ως προς τα εργαλεία που αξιοποιήθηκαν, προκύπτει το συμπέρασμα ότι, το μεν RoadRunner είναι επαρκές για την μοντελοποίηση του φυσικού περιβάλλοντος και την περιγραφή και αναπαραγωγή γεγονότων σε αυτό, στο μέτρο βέβαια που απαιτεί η προσομοίωση, ο δε CARLA είναι επίσης επαρκής και μάλιστα πληρέστερος ως προσομοιωτής, ωστόσο υστερεί στην συμμόρφωση με το ευρέως αποδεκτό πρότυπο ASAM OpenSCENARIO. Ειδικότερα, το RoadRunner βρίσκει καλύτερη εφαρμογή στην δημιουργία απλούστερων σεναρίων, και μάλιστα έχει αρκετές ενδιαφέρουσες δυνατότητες όπως η εισαγωγή χαρτών από πραγματικά δεδομένα που αξίζουν περαιτέρω διερεύνηση, ενώ σε σενάρια με περίπλοκες κινήσεις η περιγραφή αυτών είναι εφικτή με κάποιους περιορισμούς στις δυνατότητες του ερευνητή.

Η μετάβαση ενός πειράματος από το πρώτο λογισμικό στο δεύτερο, η οποία βασίζεται στην επικοινωνία μέσω των OpenDRIVE και OpenSCENARIO, είναι δυνατή με ορισμένους αναγκαστικούς συμβιβασμούς. Αυτοί προκύπτουν από τις ελλείψεις του CARLA στην υποστήριξη του OpenSCENARIO, οι οποίες εντοπίζονται και αναφέρονται στην εργασία, καθώς και γίνεται προσπάθεια αυτοματοποίησης των απαραίτητων τροποποιήσεων. Στις περιπτώσεις μικρών σεναρίων, όπως αυτών χρησιμοποιούνται για την αξιολόγηση μεμονωμένων λειτουργιών ασφαλείας του ADS, τόσο οι ικανότητες περιγραφής του σεναρίου και πειραματισμού με τις παραμέτρους αυτού από το RoadRunner όσο και η ικανότητα αναπαραγωγής των αποτελεσμάτων στον CARLA είναι ικανοποιητικές. Σε περίπλοκα σενάρια τα αποτελέσματα είναι λιγότερο ικανοποιητικά, εν μέρει και λόγω του προτύπου OpenSCENARIO καθ' εαυτόν: η ικανότητα κωδικοποίησης σύνθετων περιστάσεων κίνησης μπορεί να μην καλύπτει τις ανάγκες της έρευνας.

Αναμφίβολα η συγγραφή σεναρίων με γραφικά εργαλεία όπως το RoadRunner είναι μία εξαιρετική μέθοδος, λόγω ευκολίας, ταχύτητας και προσαρμοστικότητας. Συνεπώς, η αποτελεσματική κωδικοποίηση και μεταφορά των έργων αυτών σε προηγμένους προσομοιωτές είναι ένας σημαντικός στόχος.

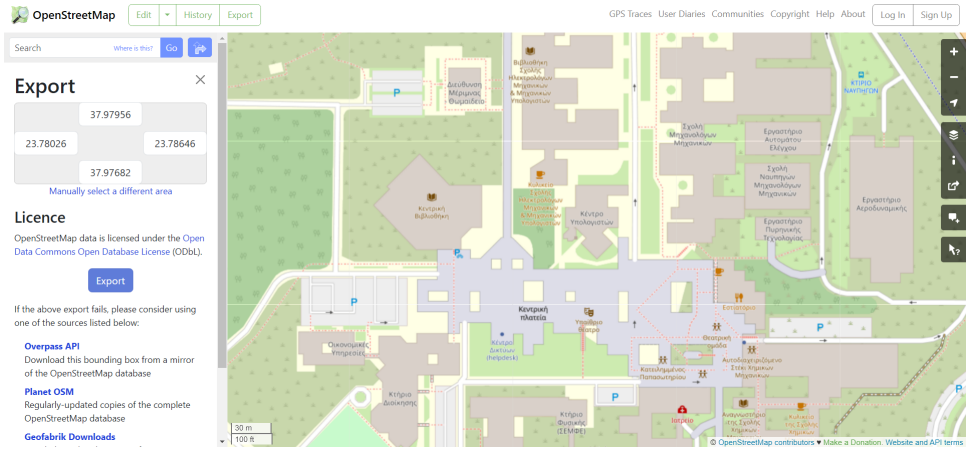
5.2 Μελλοντικές επεκτάσεις

Κατ' αρχάς η εργασία θα μπορούσε να επεκταθεί συγγράφοντας περισσότερους τύπους σεναρίων για την δοκιμασία των δυνατοτήτων και ορίων του RoadRunner και του CARLA. Πέραν αυτού, θα είχε ενδιαφέρον η αξιολόγηση και αντιπαραβολή με τον CARLA άλλων διαδεδομένων προσομοιωτών ανοικτού κώδικα, ως προς την υποστήριξη του OpenSCENARIO και την αληθοφανή αναπαραγωγή των σεναρίων. Επίσης, θα ήταν επιθυμητό να συγκριθεί η χρήση γλωσσών ειδικού σκοπού για την συγγραφή σεναρίων, όπως η SCENIC, με την χρήση των γραφικών εργαλείων όπως το RoadRunner. Τέλος, όταν η δεύτερη εκδοχή του προτύπου, η OpenSCENARIO 2.0 DSL υποστηριχθεί επαρκώς από τους προσομοιωτές, θα πρέπει να εξετασθεί για την ικανότητα περιγραφής σύνθετων σεναρίων και την ευκολία δημιουργίας παραλλαγών.

Παράρτημα

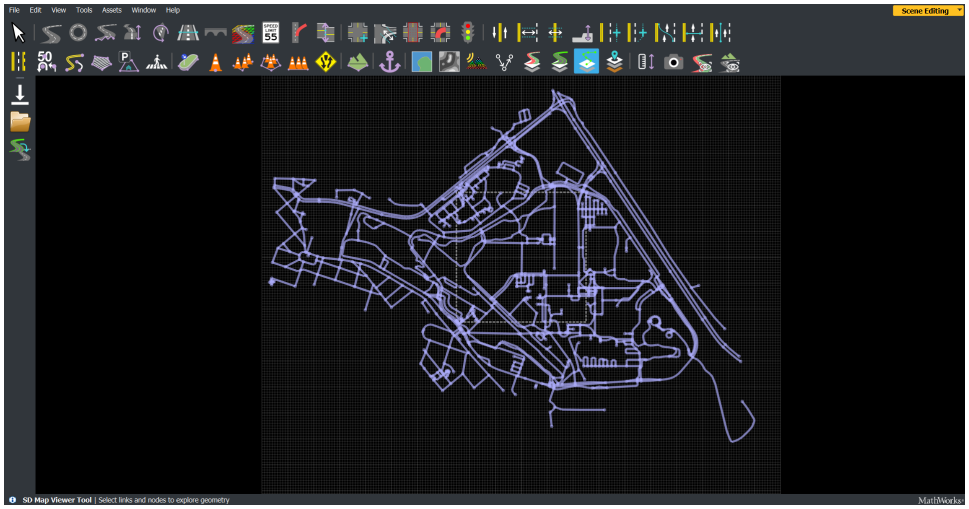
Α’: Σχεδιασμός του campus του Ε.Μ.Π. στο RoadRunner

Τα γεωγραφικά δεδομένα για την περιοχή λαμβάνονται από το OpenStreetMap, που επιτρέπει Export σε μορφή .osm.



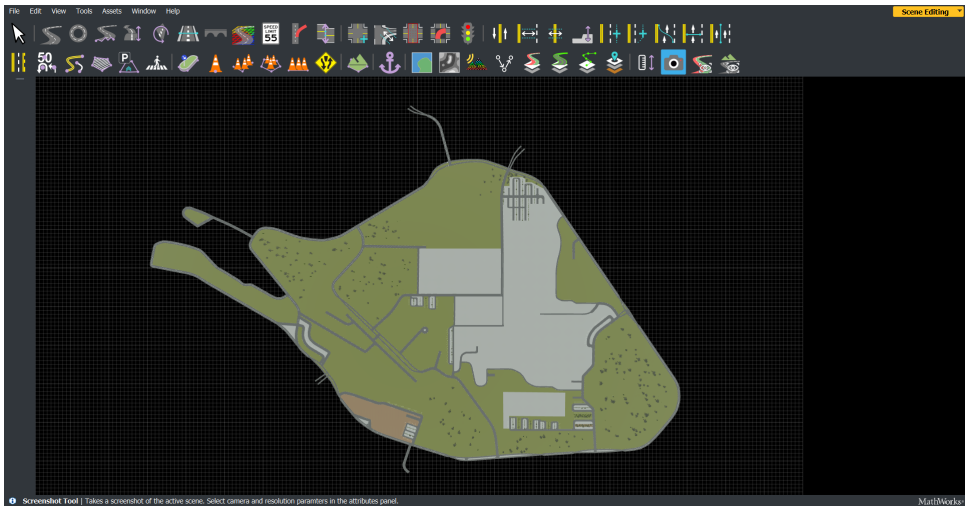
Εικόνα 39: Κάτοψη του campus στο OpenStreetMap.

Στην συνέχεια εισάγονται στο RoadRunner μέσω του SD Map Viewer Tool. Η περιοχή που καλύπτει το αρχείο είναι μεγαλύτερη από τα όρια του campus οπότε αρχικώς χρειάζεται να διαγραφούν οι περιττοί δρόμοι.



Εικόνα 40: Το επαγόμενο από το αρχείο .osm οδικό δίκτυο.

Εκτός όλων των πραγματικών δρόμων, σε αυτήν την κατάσταση αντιμετωπίζονται ως δρόμοι και οι πεζόδρομοι του campus, που επίσης χρειάζονται διαγραφή. Κατάλληλη προσαρμογή χρειάζονται και οι χώροι στάθμευσης. Στην συνέχεια ο σκελετός του οδικού δικτύου μετασχηματίζεται σε πλήρη γραφικό χάρτη με το εργαλείο Build Roads. Τελικώς προστίθενται διαφορετικές επιφάνειες ανά περιοχή και κάποια διακοσμητικά στοιχεία, ώστε να γίνεται αντιληπτή η δομή του χώρου.



Εικόνα 41: Η τελική μορφή του χάρτη του campus στο RoadRunner.

Με την μέθοδο αυτήν δεν ήταν δυνατή η μεταφορά της πληροφορίας για το υψόμετρο, με συνέπεια ολόκληρος ο χάρτης να είναι ισόπεδος. Επιπλέον δεν αποτυπώνονται τα στεγασμένα διαστήματα των δρόμων του campus.

Συμπερασματικά, η κατασκευή χαρτών στο RoadRunner με πραγματικά γεωγραφικά δεδομένα από το OpenStreetMap έχει το σημαντικότατο πλεονέκτημα της ακριβούς αποτύπωσης του σχήματος των δρόμων, ωστόσο έχει και αρκετούς περιορισμούς.

Β': Εισαγωγή χάρτη από RoadRunner στον CARLA σε Windows

Αρχική προϋπόθεση η εγκατάσταση της Unreal Engine 4 (λήψη του πηγαίου κώδικα από το repository της Epic) και του CARLA και build στον υπολογιστή.

Όλες οι ακόλουθες εντολές πρέπει να εκτελούνται από το command prompt for visual studio που εγκαθίσταται με τα Visual C++ Build Tools.

1.) Από το directory εγκατάστασης του CARLA πρώτα την εντολή `make launch` για επιβεβαίωση ότι τρέχει ο Unreal Editor.

2.) Στον Unreal Editor, Edit>Project Settings>Maps & Modes από την αριστερή στήλη για recompilation των shaders. Όταν ολοκληρωθεί κλείσιμο του Editor και

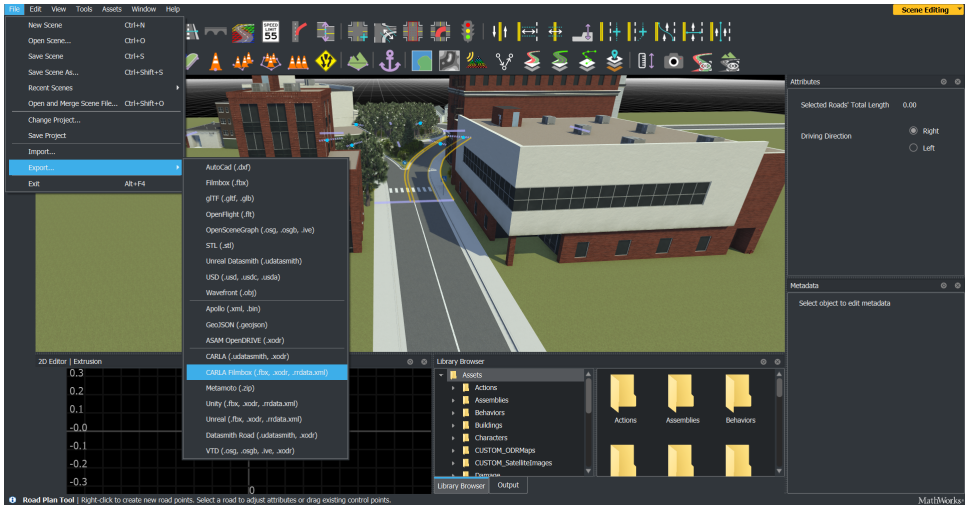
3.) `make package`

4.) το αποτέλεσμα του build θα βρίσκεται στο C:\...\carla\Build\UE4Carla\

5.) Στο αρχείο driving_simulator.conf του driving-simulator, ανάθεση στην μεταβλητή `CARLA_SIM_DIR` το ανωτέρω filepath. Σημείωση: η `CARLA_SIM_DIR` είναι το directory που θα βρίσκεται το C++ binary, ενώ η `CARLA_BIN_DIR` εκεί που θα βρίσκονται τα αρχεία python - pygame.

Έπειτα, η διαδικασία εισαγωγής χάρτη είναι η εξής:

1.) Κατ' αρχάς πρέπει να γίνει export από το RoadRunner στην κατάλληλη μορφή, την "CARLA Filmbox", ώστε να παραχθούν όλα τα απαραίτητα αρχεία. Πρέπει επίσης τα αρχεία αυτά να έχουν ίδιο όνομα με το αντίστοιχο .rrscene, το όνομα δηλαδή του χάρτη.



Εικόνα 42: Εξαγωγή χάρτη από RoadRunner.

2.) Αντιγραφή όλων των αρχείων που προκύπτουν από το RoadRunner και επικόλληση μέσα στον folder ...\carla\Import. Αν υπάρχουν μέσα σε αυτόν ήδη αρχεία από προηγούμενη εισαγωγή, πρέπει να διαγραφούν.

3.) `make import ARGS="--package=Roundabout"`

όπου "`Roundabout`" το όνομα του εκάστοτε χάρτη, πρέπει να ταυτίζεται με τα εξαχθέντα αρχεία από τον roadrunner.

4.) Τα αποτελέσματα της εντολής του βήματος 3 θα βρίσκονται στο filepath

…\carla\Unreal\CarlaUE4\Content\Roundabout

5.) Αντιγραφή του αρχείου

…\carla\Unreal\CarlaUE4\Content\Roundabout\Maps\Roundabout\Roundabout.umap

και επικόλληση στην θέση …\carla\Unreal\CarlaUE4\Content\Carla\Maps

6.) Αντιγραφή του αρχείου

…\CarlaUE4\Content\Roundabout\Maps\Roundabout\OpenDrive\Roundabout.xodr

και επικόλληση στην θέση …\CarlaUE4\Content\Carla\Maps\OpenDrive

7.) Αντιγραφή του αρχείου

…\CarlaUE4\Content\Roundabout\Maps\Roundabout\TM\Roundabout.bin

και επικόλληση στην θέση …\CarlaUE4\Content\Carla\Maps\TM

8.) Αντιγραφή του folder
…\carla\Unreal\CarlaUE4\Content\Roundabout\Static\Road\Roundabout

και επικόλληση στην θέση …\carla\Unreal\CarlaUE4\Content\Carla\Static\Road

9.) Επανάληψη του βήματος 8 για όλους τους folders στο directory

…\carla\Unreal\CarlaUE4\Content\Roundabout\Static

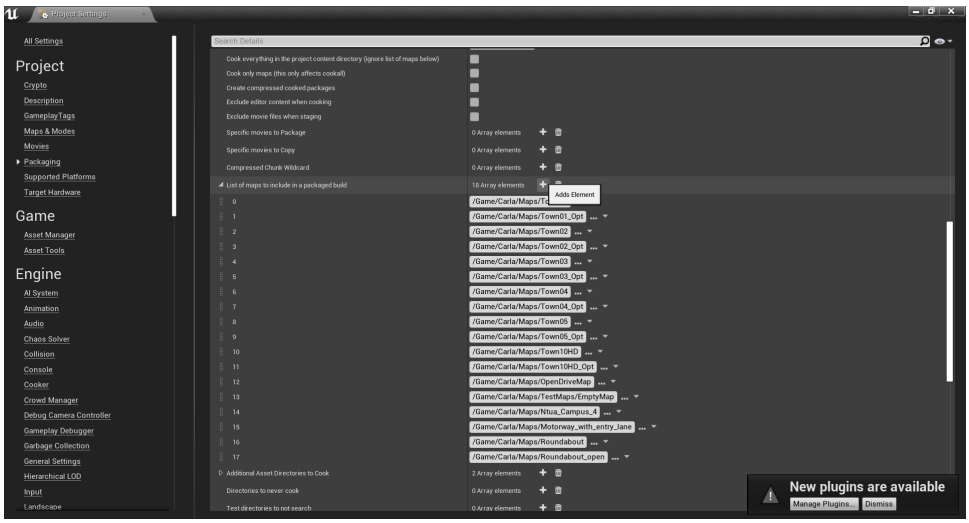
(μπορεί να περιέχει RoadLine, SideWalk, Terrain και άλλα)

και επικόλληση μέσα στους αντίστοιχους folders του directory
…\carla\Unreal\CarlaUE4\Content\Carla\Static

10.) `make launch`, και στην συνέχεια Import στο μήνυμα που θα αναδυθεί στον Unreal Editor “changes to source content files have been detected, would you like to import them?” και X στο παράθυρο “DataTable options.”

11.) Από τον Unreal editor, πρόσθεση στο Edit>Project Settings>Packaging του path για το αρχείο ...carla\Unreal\CarlaUE4\Content\Carla\Maps\Roundabout.umap στην λίστα με τους maps.

12.) `make package` στον root folder της CARLA για την δημιουργία του binary του server.



Εικόνα 43: Κατάλογος χαρτών στον Unreal Editor.

Βιβλιογραφία

[1] Norvig, Peter και Stuart Russell. *Τεχνητή Νοημοσύνη*. 2η έκδ., Αθήνα, Κλειδάριθμος, 2005.

[2] Διαμαντάρας, Κωνσταντίνος. Μηχανική Μάθηση. Αθήνα, Κλειδάριθμος, 2019.

[3] Takeda, Kazuya, et al. "A Survey of Autonomous Driving: Common Practices and Emerging Technologies." *IEEE Access, vol. 8*, 2020, pp. 58443–69. *arXiv.org*, https://doi.org/10.1109/ACCESS.2020.2983149.

[4] "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles". *SAE International*. www.sae.org/standards/content/j3016_202104/.

[5] "Terms and Definitions Related to Testing of Automated Vehicle Technologies". *SAE International*. https://www.sae.org/standards/content/dinsaespec91381/.

[6] "Mercedes-Benz Backs Redundancy for Safe Conditionally Automated Driving." *Mercedes-Benz Group*, 13 Sept. 2022, https://group.mercedes-benz.com/innovation/product-innovation/autonomous-driving/redundancy-drive-pilot.html.

[7] "Seven Mercedes-Benz Models Ready for Highly Automated and Driverless Parking." *Mercedes-Benz Group*, 13 Nov. 2023, https://group.mercedes-benz.com/innovation/product-innovation/autonomous-driving/next-step-in-driverless-parking.html.

[8] "Advanced Driving Assistance Systems & Safety Features". *BMW*. https://www.bmwusa.com/explore/driver-assistance-safety-features.html.

[9] "Official Mercedes-Benz C-Class 2022 Safety Rating". *EuroNCAP*. https://www.euroncap.com/en/results/mercedes-benz/c-class/45873.

[10] "BMW 5 Series". *EuroNCAP*. https://www.euroncap.com/en/results/bmw/5+series/50186.

[11] "Official Peugeot 308 2022 Safety Rating". *EuroNCAP*. https://www.euroncap.com/en/results/peugeot/308/45875.

[12] "ASAM OpenSCENARIO XML 1.3.0 Specification". *ASAM*. https://publications.pages.asam.net/standards/ASAM_OpenSCENARIO/ASAM_OpenSCENARIO_XML/latest/index.html.

[13] Worrall, Stewart, et al. *Automatic Lane Change Scenario Extraction and Generation of Scenarios in OpenX Format from Real-World Data*. arXiv:2203.07521, arXiv, 14 Mar. 2022. *arXiv.org*, http://arxiv.org/abs/2203.07521.

[14] Schütt, Barbara, et al. *1001 Ways of Scenario Generation for Testing of Self-Driving Cars: A Survey*. arXiv:2304.10850, arXiv, 21 Apr. 2023. *arXiv.org*, http://arxiv.org/abs/2304.10850.

[15] Heuer, Fin Malte. "Scenario Generation for Testing of Automated Driving Functions Based on Real Data". *Technische Universität Braunschweig*, 2022.

[16] “Assisted Driving. Highway and Interurban Assist Systems Test and Assessment Protocol”. *EuroNCAP*, 2024. https://www.euroncap.com/media/80151/euro-ncap-ad-test-and-assessment-protocol-v21.pdf.

[17] “Automated Driving Tests 2018”. *Euro NCAP*. https://www.euroncap.com:443/en/car-safety/safety-campaigns/2018-automated-driving-tests/. Accessed 29 Jan. 2024.

[18] “European New Car Assessment Programme, Test Protocol - AEB Car-to-Car Systems”. EuroNCAP, 2023. https://www.euroncap.com/en/car-safety/the-ratings-explained/safety-assist/aeb-car-to-car.

[19] MathWorks, *RoadRunner Scenario R2023b User Guide*. The MathWorks Inc, 2023.

[20] “ISO 34502: Test scenarios for automated driving systems. Scenario based safety evaluation framework”. *ISO*, 2022. https://www.iso.org/standard/78951.html.

[21] Gustafsson, Joakim. “Automatic Generation of Critical Driving Scenarios”. *KTH Royal Institute of Technology*, 2021.

[22] Sadeghi, Jonathan, et al. *A Step Towards Efficient Evaluation of Complex Perception Tasks in Simulation*. arXiv:2110.02739, arXiv, 4 Nov. 2021. *arXiv.org*, http://arxiv.org/abs/2110.02739.

[23] “A Framework for Automated Driving System Testable Cases and Scenarios”. *NHTSA*, 2018. https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13882-automateddrivingsystems_092618_v1a_tag.pdf.

[24] “ASAM OpenDRIVE 1.8.0 Specification”. *ASAM*. https://publications.pages.asam.net/standards/ASAM_OpenDRIVE/ASAM_OpenDRIVE_Specification/latest/specification/index.html.

[25] “ISO 19111: Geographic information. Referencing by coordinates”. *ISO*, 2019. https://www.iso.org/standard/74039.html.

[26] “ISO 8855: Road vehicles. Vehicle dynamics and road-holding ability”. *ISO*, 2011. https://www.iso.org/standard/51180.html.

[27] Lutz, Mark. *Programming Python*. 4th ed., Sebastopol, O'Reilly, 2010.

[28] “Xml.Etree.ElementTree - The Elementtree XML API”. *Python Documentation*, docs.python.org/3/library/xml.etree.elementtree.html. Accessed 3 Mar. 2024.

[29] “Unreal Engine 4 Documentation.” *Unreal Engine 4.27 Documentation*, docs.unrealengine.com/4.27/en-US/. Accessed 29 Feb. 2024.

[30] Dosovitskiy, Alexey, et al. *CARLA: An Open Urban Driving Simulator*. arXiv:1711.03938, arXiv, 10 Nov. 2017. *arXiv.org*, https://doi.org/10.48550/arXiv.1711.03938.

[31] “Carla Documentation.” *CARLA Simulator*, carla.readthedocs.io/en/latest/. Accessed 16 Dec. 2023.

[32] MathWorks, *RoadRunner R2023b Reference*. The MathWorks Inc, 2023.

[33] MathWorks, *RoadRunner R2023b User Guide*. The MathWorks Inc, 2023.

[34] MathWorks, *RoadRunner Scenario R2023b Reference*. The MathWorks Inc, 2023.

[35] “OpenStreetMap.” *OpenStreetMap*, https://www.openstreetmap.org/.

[36] Coulouris, George. *Κατανεμημένα Συστήματα Αρχές και Σχεδίαση*. Αθήνα, Da Vinci, 2020.

[37] Kleppmann, Martin. *Designing Data-Intensive Applications*. Sebastopol, O'Reilly, 2017.

[38] “Documentation”. *gRPC*, grpc.io/docs/. Accessed 29 Feb. 2024.

[39] Bokare, P. “Acceleration-Deceleration Behaviour of Various Vehicle Types”. *Transportation Research Procedia 25* (2017), p. 4733-4749. doi.org/10.1016/j.trpro.2017.05.486.

[40] “Cut-in Scenario”. *Euro NCAP*. https://euroncap.newsmarket.com/images-and-videos/video/euro-ncap---cut-in-scenario/a/4a601779-3a1c-48ae-a6a9-a0b11f1c186c.

[41] Izquierdo, R., et al. “The PREVENTION Dataset: A Novel Benchmark for Prediction of Vehicles Intentions”. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, p. 3114-21. *IEEE Xplore*, https://doi.org/10.1109/ITSC.2019.8917433.

[42] “About”. *Events Project*, 6 Mar. 2024, www.events-project.eu/about/.

[43] “Roundabouts”. *Federal Highway Administration*. https://highways.dot.gov/safety/proven-safety-countermeasures/roundabouts. Accessed 9 Mar. 2024.

[44] Ν. 2696/1999 (ΦΕΚ 57/Α` 23.3.1999). «Κύρωση του Κώδικα Οδικής Κυκλοφορίας». https://www.elinyae.gr/sites/default/files/2019-07/57A_99.pdf.

[45] “Project Jupyter Documentation”. *Project Jupyter Documentation - Jupyter Documentation 4.1.1 Alpha Documentation*, docs.jupyter.org/en/latest/.

[46] *History*, www.asam.net/about-asam/history/. Accessed 2 Feb. 2024.

[47] Fox, Armando και David Patterson. *Τεχνολογία Ανάπτυξης Λογισμικού ως Υπηρεσία*. Αθήνα, Κλειδάριθμος, 2017.

[48] Robert Bosh GmbH. *Bosch Automotive Handbook*. Wiley, 2022.

[49] Antonopoulos, Markos, et al. *Towards Collective Perception Hybrid Testing in a Roundabout Scenario with AVs*. 2024. https://doi.org/10.13140/RG.2.2.12852.42886.

[50] Bolovinou, Anastasia, et al. *Collective Perception Virtual Safety Validation in Urban Environments: Scenarios, Tools, Metrics*. 2024.

[51] Κανέλλος, Αναστάσιος. «Σύστημα οπτικής αναγνώρισης οδηγικού περιβάλλοντος για αποφυγή σύγκρουσης σε αυτοματοποιημένο όχημα επιπέδου 4». *ΕΜΠ*. 2023.

[52] Brewerton, Simon, et al. “Systems Approach to Creating Test Scenarios for Automated Driving Systems.” *Reliability Engineering & System Safety*, vol. 215, Nov. 2021, p. 107610. https://doi.org/10.1016/j.ress.2021.107610.