МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра Програмної інженерії

ЗВІТ

З НАУКОВО-ДОСЛІДНОЇ ПРАКТИКИ

Місце проходження практики «ІРЕ НАНУ»

у період з "22" січня 2018 р. по "14" квітня 2018 р.

Тема індивідуального завдання:

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

|  |  |
| --- | --- |
| Студент ПЗСм-16-1  Шпетний Д.В.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) | Керівник практики  ст. викл. Трощило О.С.  Рекомендована оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018р. |

Керівник атестаційної роботи: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

доц. Турута О.П. (оцінка)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

Комісія з прийому практики: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ проф. Четвериков Г.Г.

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ доц. Турута О.П.

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ доц. Ревенчук І.А.

(підпис)

Харків 2018

РЕФЕРАТ

Звіт з науково-дослідної практики: 21 сторінки, 3 розділів, 3 рисунків, 10 джерел.

Метою проходження практики є аналіз предметної області, аналіз можливостей сучасних технологій автономного керування транспортним засобом, постановка задачі та аналіз технологій розробки програмного комплексу для моделювання і дослідження дорожньо-транспортного середовища у контексті дослідження можливостей автономного керування.

Обрані методи розробки базуються на технологіях C#, Unity, Python.

У результат проходження практики було проведено аналіз предметної області та існуючих систем, порівняні та обрані технології розробки програмного забезпечення.

МОДЕЛЮВАННЯ СЕРЕДОВИЩА, АВТОНОМНИЙ ТРАНСПОРТ, МАШИННЕ НАВЧАННЯ, МАШИННИЙ ЗІР.

Report on research practice: 21 pages, 3 sections, 3 figures, 10 sources.

The aim of the practice is to analyze the subject area, analyze autonomous vehicle systems, formulate the problem and analyze the technologies of developing a software system for modeling and analysis of a urban road environment for research of autonomous driving.

The development methods are based on C#, Unity, Python.

As a result of the course of practice, the subject area and autonomous driving systems were analyzed, the software development technologies were compared and selected.

ENVIRONMENT MODELING, AUTONOMOUS VEHICHLE, MACHINE LEARNING, COMPUTER VISION.

ЗМІСТ

|  |  |
| --- | --- |
| Вступ | 4 |
| 1 Аналіз предметної області та постановка задачі | 6 |
| 1.1 Аналіз предметної області розробки | 6 |
| 1.2 Аналіз можливостей існуючих систем | 8 |
| 1.3 Постановка задачі |  |
| 2 Перелік вимог до програмної системи |  |
| 2.1 Призначення розробки |  |
| 2.2 Вимоги до програмного продукту |  |
| 2.3 Вимоги до клієнта |  |
| 3 UML моделювання та вибір технологій розробки |  |
| 3.1 UML – моделювання програмної системи |  |
| 3.2 Аналіз та вибір технологій розробки |  |
| Висновки |  |
| Перелік посилань |  |

ВСТУП

Людство завжди прагне до автоматизації повсякденних завдань. Користування індивідуальними засобами транспорту для багатьох є необхідністю. Розробники автомобілів ставлять за одну з найважливіших цілей підвищення комфорту та безпеки керування. Для цього постійно ведуться роботи над покращенням систем контролю та створення автоматичних допоміжних систем для водія (адаптивне керування, режими круїзної подорожі та інше).

Наступним етапом автоматизації стає усунення водія як основного центру виконування механічних дій для керування автомобілем і перетворення його ролі на контролюючу та спрямовуючу – задання маршруту, обрання типу пересування (пріоритет швидкості, комфорту, безпеки тощо) та вибір інших параметрів вищого рівня абстракції. Над проектом автономного керування працюють усі визначні автомобільні компанії (Toyota, Ford, Volkswagen, Tesla Motors) та гіганти розробки програмного забезпечення (Google, Samsung, NVidia). Однак результати їх роботи не можуть бути використаними у серійних автомобілях зараз у зв’язку з юридичними складностями та неповноцінністю систем.

Основні проблеми у дослідженні автономного пересування у мережах міста є безпосередньо неможливість навчати і тестувати систему фізично у самому місті. Підготовка та операція навіть над єдиним екземпляром автомобілю потребує значних кошт та кваліфікацій у суміжних областях діяльності, що є неможливим для невеликих груп дослідників. Один екземпляр автомобілю надає недостатньо даних для збору та обробки необхідних для покриття усіх крайових сценаріїв. Це стає очевидним навіть при типово алгоритмічному підході до задачі, не кажучи про більш вимогливу до кількості даних концепцію машинного навчання.

На кафедрі програмної інженерії університету постійно ведуться дослідження у галузях машинного зору, штучного інтелекту та моделювання складних мультиагентних систем, що дає достатню теоретичну базу для виконання даного дослідження.

Метою роботи є дослідження існуючих систем автономного пересування, транспорту та прийняття рішень задля покращення існуючих способів аналізу дорожньо-транспортної ситуації у системах автономного автомобілю завдяки використання додаткових груп сенсорів та новаторських підходів у аналізі сукупності даних. Метою практики є дослідження оптимального середовища та технологій для подальшої роботи над обраною метою дослідження.

Інтеграція систем автоматичного керування є небезпечною у першу чергу завдяки тому що неможливо передбачити та опрацювати усі сценарії поведінки звичним алгоритмічним шляхом, тому є раціональним використовувати деякі практики машинного навчання та впровадження базових концепцій штучного інтелекту, які сформують модель поведінки, що емулює дії професійного водія.

Об’єктом дослідження є автономний автомобіль та створена програмна симуляція обраних транспортних ситуацій. У ході дослідження були використані емпіричні, експериментальні методи дослідження у сукупності з абстрактно-формальним моделюванням середовища. Отримані результати удосконалюють роботу аналогічних систем контролю та аналізу середовища згідно обраним формальним критеріям. Дані наукові інновації можуть бути використаними для поліпшення існуючих систем аналізу та прийняття рішень у автоматизації керування, або інтегровані у автомобільні допоміжні підсистеми контролю та аналізу руху.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області розробки

Автономний автомобіль складається з безпосередньо транспортного засобу, який виконує роль основної платформи, комплексу сенсорів різного типу та програмної системи, що виконує роль агрегатору даних та приймає рішення на базі наявних даних [1]. Приклад такого транспортного комплексу наведений у схематичному вигляді на рисунку 1.1.



Рисунок 1.1 – Схематичний приклад автономного автомобілю

Традиційним підходом до вибору сенсорів є наявність камер та візуальних сенсорів різного формату, що надають недостовірні дані по окремості, але їх сукупність дозволяє відтворити більш дійсний склад речей. Розглянемо наявні сенсори та їх можливості у контексті аналізу дорожньо-транспортної ситуації.

Предметна область розробки знаходиться на перетині галузей робототехніки, прийняття рішення, комп’ютерного зору, машинного навчання. У даному дослідженні зроблений акцент на програмних системах, тому деякі деталі з інших галузей подані у спрощеному вигляді або не розглядаються взагалі.

Типовий склад сенсорів у автономному транспортному засобі [1] поданий на рисунку 1.2 у схематичному форматі.

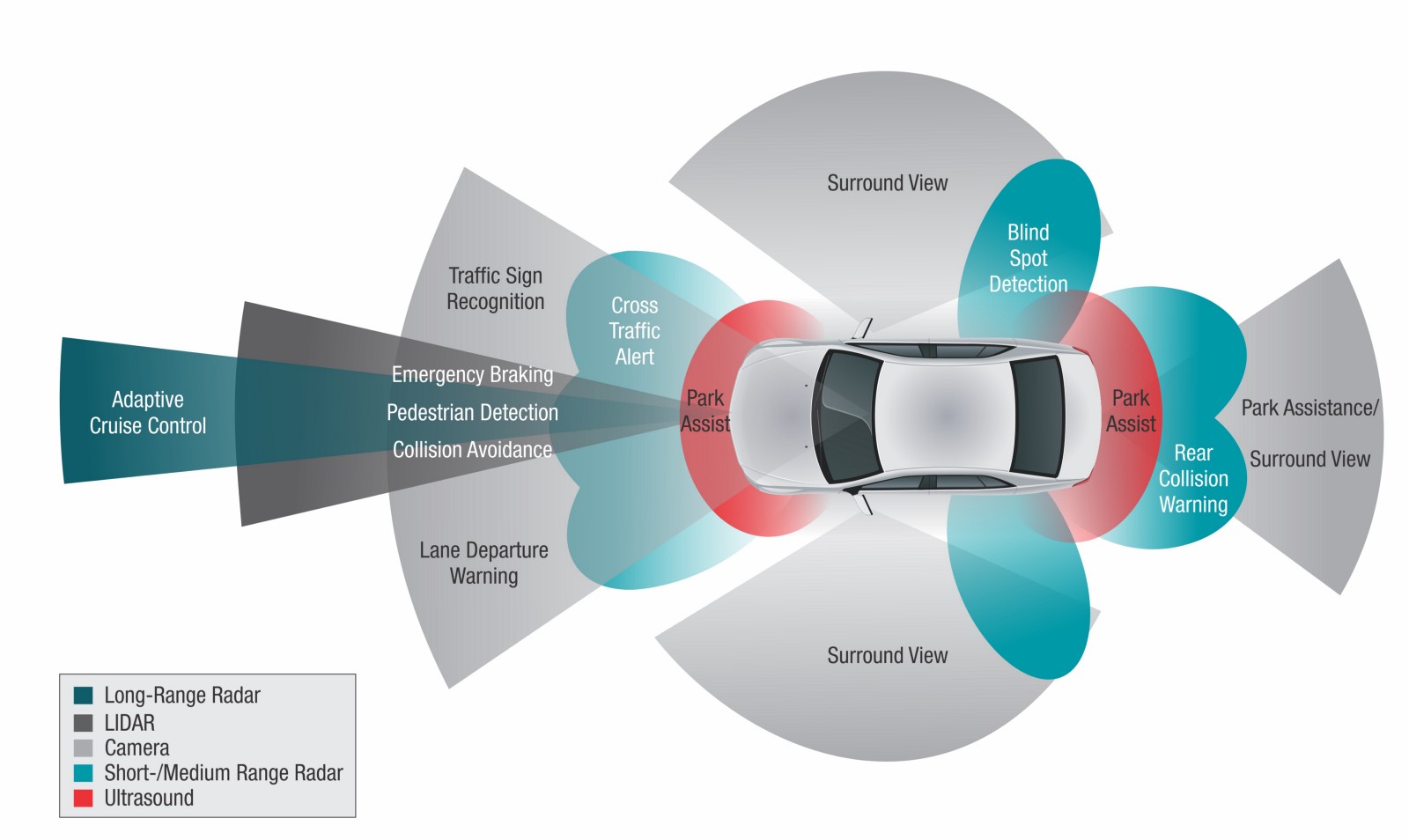


Рисунок 1.2 – Детектори автономного автомобіля

Загалом, можна виділити основні класи проблем, що вирішує кожен с датчиків:

* критичні ситуації – LIDAR, камера, допоміжні радари короткого радіусу дії. До таких ситуацій переважно входять виявлення аномальної поведінки пішоходів або інших динамічних агентів на дорозі таких як автомобілів, тварин, падаючих гілок чи каменів.
* допоміжні інформаційні дані – камери, радари різного радіусу дії, ультразвукові радари. До цього класу можна віднести аналіз типових сліпих зон водія, допомога у аналізі дорожньої ситуації (знаки, світлофори, дорожня розмітка), допомога у парковці.
* системи покращення життя водія – камера, радар дальнього радіусу дії. Переважно це такі системи як круїз контроль, моделювання навколишнього середовища.

LIDAR (Light Identification, Detection and Ranging) – це технологія отримання та обробки інформації про віддалені об'єкти за допомогою активних оптичних систем, що використовують явища відбиття світла і його розсіювання в прозорих і напівпрозорих середовищах. Скануючі лідари в системах машинного зору формують двовимірну або тривимірну картину навколишнього простору [2].

Ключовими з точки зору проектування автономного автомобілю є питання використання даних з сенсорів у реальному-часі, що накладає певні обмеження до складності та оптимізації обчислень. Одними з систем такого роду є комп’ютерні ігри, які можуть складати базу оточення для дослідницьких робот з візуальним зображенням [3].

Окрім безпосередньо потоку інформації також необхідно обробляти його для прийняття рішень у керуванні транспортним засобом, чим займається підсистема аналізу та прийняття рішень.

На сьогодні існує дві основні парадигми для автономних систем водіння на основі комп'ютерного зору: опосередкований підхід до сприйняття (mediated perception approach) щоб розібрати всю дорожню сцену, і підхід рефлекторного аналізу поведінки для прийняти рішення про водіння (behavior reflex approach) яка складає пряму залежність між зображенням та безпосередньою дією завдяки регрессору.

Mediated perception approach складається з багатьох підкомпонентів для розпізнавання об’єктів що відносяться до водіння таких як дорожня розмітка, знаки, лінії руху, світлофори, автомобілі, пішоходи тощо. Результати аналізу складаються у загальну картину безпосереднього оточення автомобілю. Для контролю транспортного засобу система штучного інтелекту бере до уваги всю інформацію перед прийняттям рішення. Оскільки тільки невелика кількість об’єктів має значення у виборі рішення [4], то рівень повного аналізу оточення може додати непотрібний шар складності до і так важкої задачі. На відміну від багатьох задач робототехніки автономний автомобіль контролює лише два параметри – швидкість та напрямок. Остаточний простір рішень має невеликий ступінь свободи на порівняння зі складністю розмірності моделі оточуючого світу. Повнота інформації також надає необхідність використовувати велику кількість сенсорів, що в свою чергу підвищує загальну ціну системи.

Behavior reflex approach складається з прямої залежності між сенсорними вхідними даними та безпосередньо дією автомобіля. Ідея полягає у використанні нейронної мережі для створення прямого мапінгу від зображення до дії. Для навчання використовується реальний водій. Система зберігає зображення та дії водія під час конкретного зображення. Така концепція елегантна у своїй простоті, але вона може зазнавати нижче приведених складностей під час пересування у реальному трафіку та під час виконання складних водійських маневрів. По-перше, водії приймають різні рішення під час однакових ситуацій, що може спричинити розбіжності під час навчання регрессора. Наприклад, якщо автомобіль знаходиться прямо на шляху, то один водій може продовжувати рухатися за машиною, інший – обігнати зліва або справа. Коли всі сценарії присутні у даних, то модель зазнає складностей у прийнятті рішення хоча сенсорні початкові дані однакові. По-друге, система прийняття рішень на такій базі є занадто низькорівневою. Прямий мапінг не може розібратися з повною картиною ситуації. Для прикладу, з точки зору такої моделі обгін автомобілю та пересування назад на полосу є низкою дуже низькорівневих рішень як поворот руля на певний час, а потім поворот у інший бік на певний час. Система не може правильно реагувати у критичних ситуаціях та не може відслідковувати реальний стан речей, які підходять для водіння – необхідний рівень абстракції – це маневр обгону або пересування за автомобілем, а не ступінь повороту руля.

У даній роботі розглядається така модель, що поєднує у собі ці дві концепції. Ця парадигма займає місце між першими двома. Ключовою ідеєю концепції є існування певних станів та індикаторів доступності (affordance indicator) у контексті дорожньої ситуації [5]. Індикатори можуть обиратися в залежності від обраної задачі. Наприклад, дистанція до інших автомобілей, дистанція до дорожньої розмітки тощо.

Важливим компонентом у автономному автомобілі є підхід до штучного інтелекту, який приймає рішення на основі даних. Концептуально можна поділити системи штучного інтелекту на ті що навчаються з учителем (supervised learning) та на самонавчаємі (reinforcement learning).

Традиційним підходом для систем автономного транспорту є використання даних отриманих під контролем людського водія. Це надає змогу отримувати лише коректні дані та спростити кількість необхідного навчання у силу оптимальності контролю автомобіля досвідченим водієм. Недоліками концепції є неможливість моделювання екстремальних ситуацій та можливі протиріччя на великих даних, оскільки кожен водій унікально реагує на дорожню ситуацію. Також процес збору даних накладає потенційні обмеження, оскільки людина вирішує що саме є релевантним у процесі збору вхідних даних.

Навчання з підкріпленням на відміну від першого підходу не потребує людського водія та початкових даних. Основою концепції є встановлення цілей для досягнення та критерії оптимальності їх виконання. Наприклад, для автомобіля це може бути подорож із точки А в точку Б з мінімальною кількістю збитих пішоходів, або з найменшим рівнем витрати пального. Часто використовують комбінацію критеріїв для оптимізації сценаріїв поведінки. Недоліками такого підходу є значно більший час на навчання, складність системи, необхідність проведення ретельного аналізу критеріїв оптимальності.

Згідно з останніми дослідженнями саме навчання з підкріпленням дає найбільш оптимальні результати у задачах з неможливістю передбачити всі сценарії події та з необхідністю реагувати у задане вікно часу [7]. Наприклад, одним з важливих досягнень у цей сфері є перемога AlphaGo над людиною у старовинній грі го. Її особливість саме у непередбачуваності ситуації та у неможливості прорахувати всі варіанти розвитку гри. Впровадження такої системи штучного інтелекту потенційно надає можливість виправити недоліки зазначені у попередньому пункті. Приклад структури системи навчання з підкріпленням наведений на рисунку 1.3 у схематичному форматі.

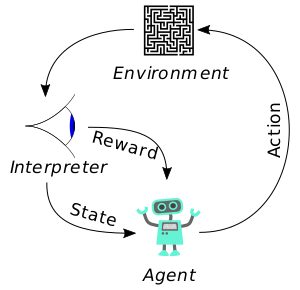


Рисунок 1.3 – Модель reinforcement learning

У області комп’ютерного зору дослідники вивчали кожну підзадачу окремо [8]. Розпізнавання машин та окремих полос руху є основою моделі автономного автомобілю. Типові алгоритми на виході видають обмежувальні геометричні кордони (bounding boxes) машин та лінії (spline) розмежувальних полос.

Тобто для області дослідження автономного руху необхідно комплексно підходити до проблему з урахуванням наведених факторів ризику та недоліків попередніх систем і досліджень. Одним з основних недоліків у даній сфері діяльності безумовно є фінансова складова та неможливість проводити експерименти у реальному світі. Автомобілі коштують дорого, обладнувати навіть один такий усіма необхідними датчиками є неможливим для невеликої дослідницької організації.

В результаті аналізу предметної області було вирішено провести дослідження концептуальних підходів до безпосередньо реалізації автономного автомобілю та систем його навчання на основі комп’ютерної моделі.

1.2 Аналіз можливостей існуючих систем

У даний час існує велика кількість досліджень автономних автомобілей та пов’язаних суміжних галузей, але більшість з них не є комплексними системами або є частиною комерційної таємниці, оскільки перспективи досліджень та потрібні економічні інвестиції недоступні невеликим дослідницьким організаціям. Далі детально розглянемо та проаналізуємо системи моделювання оточуючої середи для створення моделі автономного пересування, обрані для дослідження.

CARLA – це симулятор взаємодії машини та оточуючої середи з відкритим кодом. Розроблений спеціально для досліджень автономного автомобіля з урахуванням розробки, навчання, валідації в умовах міського пересування. Екранні форми симулятора наведені на рисунку 1.4 за різних умов симуляції.



Рисунок 1.4 – Симулятор CARLA

Платформа надає деяку кількість візуальних ресурсів, реалізацію частини наведених у розділі аналізу предметної області сенсорів. Але дана платформа призначена для більш масштабних симуляцій тому є з значним ускладненням на порівнянні з необхідними функціями. Також стек технологій розробки накладає певні складності у роботі з симулятором.

GTA V – як не здається дивним, але саме сучасні комп'ютерні ігри є ідеальним середовищем для дослідження деяких систем автономного керування. Приклад фронтальної камери на автомобілі у грі наведений на рисунку 1.5, такий варіант можна застосовувати для тренування візуальної частини регресора.



Рисунок 1.5 – Екранна форма процесу водіння у GTA V

Така система середовища дозволяє вивчати процес керування у форматі чорного ящику, а складність гри знімає необхідність додатково моделювати поведінку середовища (пішоходи, транспорт). Недоліком безумовно є відсутність коду гри та неможливість модифікувати початкові умови за межами гри. Також під питанням є використання гри у академічних цілях довготривалого характеру.

Згідно виявлених недоліків існуючих симуляцій було вирішено розглядати розробку власної системи симуляції, яка вдовольняла б потребам саме нашої дослідницької кафедри.

1.3 Постановка задачі

Метою проходження науково-дослідницької практики є проведення аналізу предметної області, аналіз підходів та концепцій у дослідженні автономного автомобілю, виявлення можливостей та недоліків віртуальних моделей урбаністичного оточення, вибір технологій розробки моделі та загального стеку технологій на базі якого буде проводитися подальше дослідження, UML-моделювання системи. Дана система буде реалізована у якості атестаційної роботи магістра.

Після проведення досліджень програмний комплекс має вирішувати наступні проблеми та мати такі властивості:

* легкість роботи с системою емуляції міського середовища;
* модульна архітектура, яка спростить додання нових елементів;
* складатися з безкоштовних елементів або бути безкоштовною для використання у академічному середовищі;
* мати тісний зв’язок з технологіями, що вивчаються кафедрою програмної інженерії.

Прототип повинен мати наступні властивості:

* відтворювати частину урбаністичного середовища;
* мати дорогу з коректною розміткою;
* мати керуємий світлофор;
* мати пішохідний перехід та модель пішохода;
* моделювати поведінку реального автомобілю з урахуванням основних його фізичних властивостей (маса, прискорення, коефіцієнт тертя з дорогою та інше);
* мати низку камер стандартного формату;
* можливість зберігати зображення з камер у автоматичному режимі.

У результаті буде отриманий програмний комплекс, який буде використаний у якості платформи для подальших досліджень кафедрою програмної інженерії.

2 ПЕРЕЛІК ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

2.1 Призначення розробки

Перед початком роботи над програмною системою необхідно визначитися з вимогами до неї. У загальному випадку під вимогами розуміють сукупність властивостей, які повинна мати система, що реалізується. У ході роботи треба реалізувати програмну систему для симуляції урбаністичного транспортного середовища.

Даний додаток призначений для дослідників проблеми автономних автомобілей, проблем машинного зору та може бути взятий як платформа для проведення деяких фізичних симуляцій. Оскільки дана розробка є базою саме для дослідження методів машинного навчання у контексті автономних автомобілей, то також програмний комплекс повинен взаємодіяти з існуючими системами машинного навчання та популярними технологіями для їх реалізації.

2.2 Вимоги до програмного продукту

Загальні відомості про розробку системи полягають у наступному: система емуляції урбаністичного середовища має бути розроблена за допомогою мови програмування C# на базі Unity. Це дозволить забезпечити модульність архітектури, простоту роботи із додатком та дозволить запускати симуляцію на різних операційних системах. Симуляція повинна обробляти дані з контролеру (за умови керування автомобілем людиною) або мати можливість емулювати керування автомобілем за умовою надання дій штучним інтелектом. Дані з симуляції повинні оброблятися на всіх платформах та мати структурований формат якщо необхідно буде додатково обробляти їх. У результаті симуляція повинна забезпечувати відносну швидкодію роботи з нижньою границею не нижче 30 кадрів за секунду за умови віконного режиму роботи з розмірами 800x600 пікселей.

Система розробляється у вигляді окремого додатку який буде компілюватися окремо під кожну необхідну платформу. Однак до клієнту є окремі вимоги – наявність окремого графічного прискорювача (відеокарти) та останніх версій драйверів. Підтримуються платформи Windows, Unix (Linux), OS X та смартфонні операційні системи Android 4.4.x та вище і iOS версії 8 та вище.

Система машинного навчання повинна бути реалізована на базі мови Python з використанням необхідних фреймворків та бібліотек, таких як OpenCV, TensorFlow тощо. Використання готового функціоналу дозволить зосередитись на дослідницькій новаторській роботі. Безпосередньо штучний інтелект повинен однаково працювати на будь-якій операційній системі з списку наведеного вище.

3 UML МОДЕЛЮВАННЯ ТА ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

3.1 UML – моделювання програмної системи

Моделювання розроблюваної системи проводиться з використанням мови UML для побудови діаграм, що допоможуть відобразити функціональність та внутрішню структуру системи. UML - мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення. UML [9] є мовою широкого профілю, це відкритий стандарт, який використовує графічні позначення для створення абстрактної моделі системи, що називається UML-моделлю.

Можна виділити наступні переваги UML:

– UML об'єктно-орієнтований, в результаті чого методи опису результатів аналізу і проектування семантично близькі до методів програмування на сучасних об'єктно-орієнтованих мовах;

– діаграми UML порівняно прості для читання після досить швидкого ознайомлення з його синтаксисом;

– UML розширює і дозволяє вводити власні текстові та графічні стереотипи.

При проектуванні системи були розроблена діаграма класів. Діаграма класів показує статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. На даному прикладі вона описує функціональне призначення підсистеми машинного навчання з тренером. Розробка діаграми має такі цілі:

– визначити архітектуру підсистеми, яка моделюється;

– сформулювати загальні вимоги до функціонального поведінки проектованої системи;

– розробити вихідну концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей.

Діаграма класів для загальної системи машинного навчання відображена на рисунку 3.1.

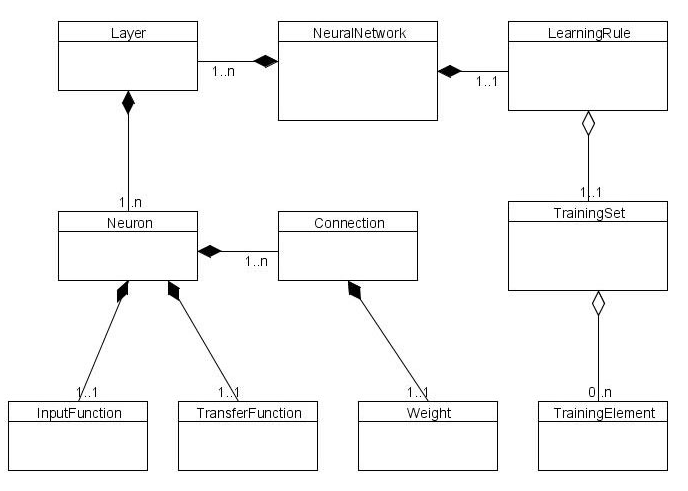


Рисунок 3.1 – Діаграма класів

Діаграма класів моделює систему з точки зору розробника і установлює абстрактні кордони високого рівня. Діаграми цього типу допомагають при аналізі вимог до системи та проектуванні архітектури високого рівня.

3.2 Аналіз та вибір технологій розробки

Симуляція була побудована за допомогою ігрового движка Unity та мови програмування C#. Даний вибір технологій обґрунтовується їх популярністю, стабільністю роботи та загальною простотою використання у контексті даної задачі. Основною метою архітектури було забезпечити модульну компонентну архітектуру. Загальна архітектура Unity наведена на рисунку 3.2.

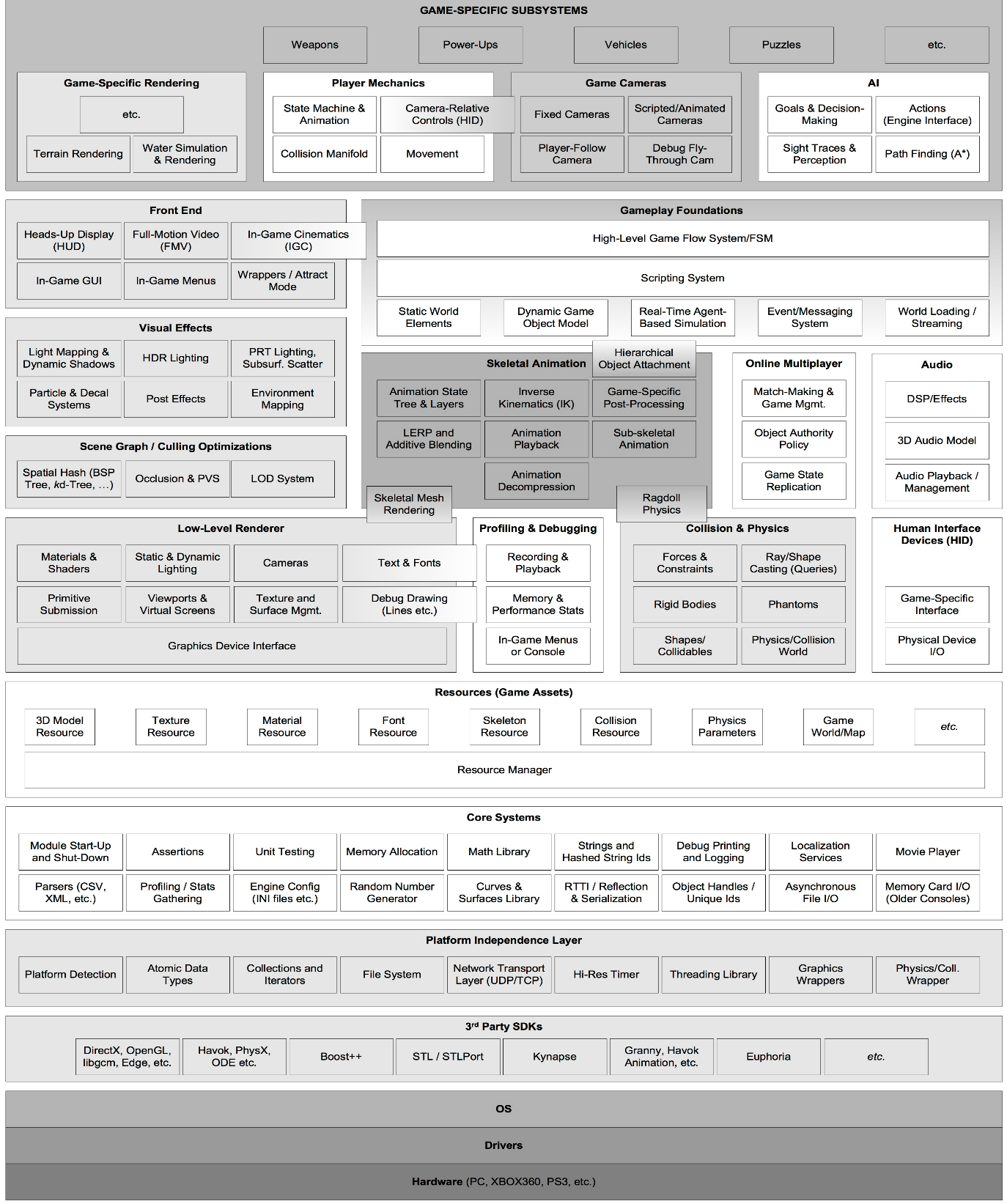


Рисунок 3.2 – Архітектура Unity

Unity – це багатоплатформовий інструмент для розробки тривимірних додатків та ігор для операційних системам Windows, OS X, .Android, iOS, Linux.

Також є можливість створювати інтернет-додатки за допомогою спеціального під'єднуваного модуля для браузера Unity, а також за допомогою експериментальної реалізації в межах модуля Adobe Flash Player. Застосування, створені за допомогою Unity, підтримують DirectX, OpenGL, Vulkan [9].

Головна мова програмування у Unity – це C#. C# - об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Синтаксис C# близький до С++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. C# розроблялась як мова програмування прикладного рівня для CLR і тому вона залежить, перш за все, від можливостей самої CLR. Це стосується, перш за все, системи типів C#. Присутність або відсутність тих або інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована у відповідні конструкції CLR. Збірка сміття не реалізована в самому C#, а проводиться CLR для програм, написаних на C# точно так, як і це робиться для програм на VB.NET, J# тощо. [10].

Однак однією з особливостей Unity є велика кількість підтримуваних платформ. Тому мова програмування всередині самого ігрового рушія не є повною копією реалізації від Microsoft [10]. Такий підхід необхідний задля коректної роботи на багатьох платформах та для можливості подальшої оптимізації ігрового коду. Подальша оптимізації можлива оскільки мова програмування створюється як інструмент широкого профілю, а у контексті ігрового рушія мова стає провідником логіки, яку реалізує безпосередньо сам рушій.

Важливою частиною у виборі технологій було питання ліцензування та ціни за використання. Даний ігровий рушій є повністю безкоштовним для некомерційних розробок, тобто є безкоштовним і для поточних академічних досліджень. Обрана сукупність інструментів є популярною для розробки ігрових додатків та для програмування у цілому, що збігається з навчальним курсом університету та інтересами студентів. Для даного дослідження це означає, що бажаючі долучитися будуть мати необхідну мінімальну кваліфікацію та зможуть знайти рішення більшості типових проблем пов’язаних саме з технологією, що у свою чергу надасть більше часу на безпосередньо дослідження.

Дослідження машинного навчання та систем контролю автономного автомобілю буде відбуватися за допомогою мови програмування та фреймфорків і біблотек Python.

Python – це інтерпретована об'єктно-орієнтована мова програмування високого рівня з строгою динамічною типізацією. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання існуючих компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована [11].

Для реалізації запланованого порівняльного дослідження парадигм машинного навчання та комп’ютерного зору заплановано використовувати бібліотеки OpenCV, ImageGrab, фреймворк машинного навчання TensorFlow. Даний набір інструментів дозволить зосередитись на абстракції більш високого рівня, а не витрачати час на реалізацію давно відомих алгоритмів, які не відносяться до цілей дослідження.

Git буде використаний для контрою версій під час розробки додатку. Git – розподілена система керування версіями файлів та спільної роботи. Git є однією з найефективніших систем керування версіями, що надає гнучкі засоби розробки. Метою проекту також є заохочування студентів до дослідницької роботи, частиною якої буде праця у команді.

ВИСНОВКИ

У ході проходження науково-дослідної практики було проведено аналіз предметної області, порівняні та обрані технології розробки системи моделювання оточуючої середи для урбаністичного автономного транспортного засобу.

На початку роботи було проведено аналіз предметної області, в процесі якого були визначені основні проблеми, що існують в розглянутій галузі. На підставі аналізу предметної області була проведена постановка завдання.

У ході виконання поставленої задачі було:

* проведено аналіз предметної області;
* проведено UML-моделювання архітектури симуляції;
* проведено аналіз та обрані технології розробки симуляції;
* проведено аналіз та обрані технології подальшої дослідницької роботи з модулем машинного навчання та комп’ютерного зору.

У результаті аналізу потреб та вимог було визначено, що кінцевий продукт повинен мати наступні властивості:

– модульна архітектура;

– простота продовження розробки;

– мультиплатформеність;

– достатню швидкодію.

Розроблений прототип, що відповідає вимогам сформованим на етапі проектування та планування задачі. Прототип дозволяє на примітивному рівні розпочинати наступний етап наукового дослідження пов’язанний з машинним навчанням та комп’ютерним зором. Також прототип відповідає архітектурним вимогам і дозволяє швидко розширювати його функціонал.

ПЕРЕЛІК ПОСИЛАНЬ

1. Вільна енциклопедія Википедия [Електронний ресурс] / Мережева енциклопедія Wikipedia, 2000. - Forefront TMG 2010: Режим доступу: : www/ URL : http://ru.wikipedia.org/ . – Загол. з екрану.

2. Фаулер, М., Дейвид, Р. Архитектура корпоративных программных приложений [Текст] / М. Фаулер, Д. Райс; – М.: Вильямс, 2008. – 544 с.

3. Фаулер, М. UML. Основи, 3-є видання: пер. з англ. [Текст] / М. Фаулер. – СПб: Символ-Плюс., 2004. – 192 с. – ISBN 5-93286-060-Х.

4. Діаграма варіантів використання (use case diagram) [Електронний ресурс] / UML Теорія – Режим доступу: http://www.info-system.ru/designing/methodology /uml/theory/use\_case\_diagram\_theory.html/ . – Загол. з екрану.

5. Duckett, J. HTML and CSS: Design and Build Websites. 1st Edition. [Текст] / J. Duckett. – John Wiley & Sons, 2011. – 490 c. – ISBN-13: 978-1118008188.

6. Флэнаган, Д. JavaScript. Подробное руководство [Текст] / Д. Флэнаган. – СПб. Символ-Плюс, 2013. – 1080 с. ISBN 978-5-93286-215-5.

7. Wahling, D. Angular in 60 Minutes [Текст ]/ D. Wahling. – Wahlin Consulting, 2014. – 102c.

8. Документація Nodejs [Електронний ресурс] / Офіційний сайт проекту Nodejs. – Режим доступу: www/ URL: https://wwwnodejs.org/ . – Загол. з екрану.

9. Уорсли, Дж., Дрейк, Дж. PostgreSQL. Для профессионалов [Текст] / Дж. Уорсли, Дж. Дрейк. – СПб: Питер, 2003. – 496 с.

10. Colborne, G. Simple and Usable Web, Mobile, and Interaction Design (Voices That Matter) 1st Edition. [Текст] / G. Colborne. – New Riders, 2010. – 208 c.