

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра інформатики

Курсова робота

освітній ступінь – бакалавр

на тему: «**Апаратно-програмне рішення комп'ютерних ігор на платформі
Arduino для людей з вадами зору**»

Виконав: студент 3-го року навчання,

Спеціальності

122-Комп'ютерні Науки

Студента Гайворонського Романа

Вячеславовича

Керівник Бабич Т.А., _____

магістр комп'ютерних наук, асистент

« ____ » _____ 20 ____ р.

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра інформатики

Освітній ступінь бакалавр

Спеціальність 122-Комп'ютерні Науки

Освітня програма бакалавр

ЗАТВЕРДЖУЮ

Завідувач кафедри інформатики

Гороховський С. С.

“ _____ ” _____ 20__ року

ЗАВДАННЯ

ДЛЯ КУРСОВОЇ РОБОТИ СТУДЕНТУ

Гайворонського Романа Вячеславовича

1. Тема роботи «**Апаратно-програмне рішення комп'ютерних ігор на платформі Arduino для людей з вадами зору**», керівник роботи Бабич Трохим Анатолійович, магістр комп'ютерних наук, асистент
2. Строк подання студентом роботи 24.05
3. План роботи

Розділ 1. Дослідження та аналіз предметної області

- 1.1. Загальна інформація
- 1.2. Проблематика
- 1.3. Ехолокація
- 1.4. Ray casting

Розділ 2. Огляд використаних інструментів

- 2.1. Мікроконтролер
- 2.2. Ігровий рушій та комунікаційний інтерфейс

Розділ 3. Проектування та розробка системи

3.1. Апаратна складова

3.2. 3D модель

3.3. Базова відеогра Картинг

3.4. Розширення Ardity

3.5. Протокол комунікації

3.6. Реалізація ехолокації

3.7. Тактильна віддача

ГРАФІК ПІДГОТОВКИ КУРСОВОЇ РОБОТИ ДО ЗАХИСТУ

№ з/п	ПЕРЕЛІК РОБІТ	Термін виконання	Дата ознайомлення наукового керівника	Підпис наукового керівника	Примітки
1.	Вибір теми, затвердження її на засіданні кафедри та закріплення наукового керівника Узгодження календарного графіка підготовки кваліфікаційної роботи. Ознайомлення студента з критеріями оцінювання кваліфікаційної роботи (п. 8.5).	10 жовтня 2020			
2.	Вивчення джерел літератури, матеріалів архівів, періодичних видань, збір та узагальнення фактів, даних	10 жовтня 2020 – 2 листопада 2020			
3.	Складання плану каліф. роботи та узгодження з науковим керівником	2 листопада 2020			
4.	Написання розділів роботи	2 листопада 2020 – 01 березня 2021			
5.	Проміжний контроль виконання роботи	01 лютого 2021			
6.	Написання кваліфікаційної роботи в цілому, ознайомлення з її першим варіантом наукового керівника	11 січня 2021 – 29 березня 2021			
	Розділ 1 (постановка проблеми, теоретичні основи, огляд літературних джерел)	25 січня 2021			
	Розділ 2 (аналітично-дослідницька частина)	01 березня 2021			
	Розділ 3 (проектно-рекомендаційна частина)	29 березня 2021			
7.	Повне завершення написання кваліфікаційної роботи, оформлення її згідно з вимогами й подання на відгук науковому керівнику	01 квітня 2021 – 06 травня 2021			
8.	Подання кваліфікаційної роботи для перевірки письмових робіт студентів НаУКМА на відповідність вимогам академічної доброчесності,	17 травня 2021			
9.	Публічний захист кваліфікаційної роботи перед екзаменаційною комісією	згідно з розкладом роботи ЕК			

Графік узгоджено 10 жовтня 2020 р.

Науковий керівник Бабич Трохим Анатолійович

Виконавець курсової роботи Гайворонський Роман

Зміст

Анотація	1
Вступ	2
Розділ 1. Дослідження та аналіз предметної області	3
1.1. Загальна інформація	3
1.2. Проблематика	4
1.3. Ехолокація	5
1.4. Ray casting	6
Розділ 2. Огляд використаних інструментів та технологій	7
2.1. Мікроконтролер	7
2.2. Ігровий рушій та комунікаційний інтерфейс	8
Розділ 3. Проектування та розробка системи	11
3.1. Апаратна складова	11
3.2. Прототипування	11
3.3. Основа проекту	12
3.5. Протокол обміну даних	14
3.6. Метод ехолокації	15
3.7. Використання тактильної інформації	16
Висновки	18
Список використаних джерел	19
Додатки	20

Анотація

В даній роботі розглянуто апаратно-програмне рішення комп'ютерних ігор на платформі Arduino для людей з вадами зору. Метою роботи було надати нові інструменти розробки відеоігор за допомогою почуття дотику. Отримані необхідні результати і зроблено висновки.

Вступ

На сьогодні комп'ютерні ігри є невід'ємною частиною індустрії розваг. В останні роки розробники ігор витрачають велику кількість часу і ресурсів на адаптацію під людей з особливими потребами, а саме: для людей з дальтонізмом зазвичай передбачено режим дальтоніка, для людей з поганим слухом - субтитри і/або текстові підказки. Для слабкозорих багато розробників впроваджують озвучення користувацького інтерфейсу[1].

Однак як бути з людьми, в яких наявна вроджена або набута сліпота: для таких людей зазвичай розробляють застосунки з максимально практичним призначенням. Наприклад, застосунок віртуальної реальності для розвитку навичок користування тростиною.

Актуальним питанням залишається специфіка розробки саме розважальних ігор. Адже сліпі гравці не здатні користуватися основним інструментом передачі інформації, - графічним інтерфейсом, - як зрячі. А значить, при роботі залишається зосередитися лише на чуттях слуху і дотику гравців. Однак дотиком нерідко нехтують, існують портали і форуми, де люди створюють ігри [2], базуючись лише на звуковій складовій. Метою даної роботи є розширення концепції, та інкорпорування в ігровий процес і відчуття дотику, для більш ефективною передачі інформації гравцю. Для цього був використаний мікроконтролер Arduino з вібромоторами, а також інтерфейс для передачі даних з ігрового рушія Unity до мікроконтролера. Таким чином, нам відкривається цілий спектр нових можливостей для донесення інформації гравцеві.

Розділ 1. Дослідження та аналіз предметної області

1.1. Загальна інформація

Ринок контролерів і аксесуарів для людей з різними захворюваннями, які не здатні використовувати звичайні пристрої для управління смартфонами / комп'ютерами / консолями досить широкий. Велика кількість компаній і стартапів пропонують свої, унікальні, рішення. Наприклад контролер QuadStick [3] для людей з тетраплегією (м'язовою дистрофією) дає можливість грати в ігри та управляти курсором з клавіатурою використовуючи лише рот, або спеціальні кісткові слухові апарати для передачі звуку безпосередньо користувачеві у середнє вухо за допомогою вібрацій, для людей з поганим слухом. Однак, такі аксесуари досить дорогі, а також, на жаль, поки що не існує пристроїв для передачі цифрової картини слабо зрячим людям.

В останнє десятиліття були проведені наукові роботи в даному полі, що досліджують вплив аудіо і тактильної інформації на ефективність навчання когнітивним моторним навичкам у слабо зрячих людей. Джейме Санчез з Центру Просунутих Досліджень в Навчанні в своїй роботі з'ясував: щодо зручності використання, як гаптичний(тактильний), так і звуковий інтерфейси були ефективними для забезпечення адекватного зворотного зв'язку з користувачем у віртуальному середовищі. Використання тактильного та аудіо інтерфейсів разом дозволили сліпому користувачеві, який здійснював навігацію по віртуальному середовищу гри, мати змогу сформуванати краще сприйняття відстаней, форм та орієнтації предметів на карті при оновленні своєї позиції.

Також користувачі розвили свої орієнтаційні та рухові навички в результаті взаємодії з відеоіграми, що розробила команда, які

безпосередньо пов'язані з ефективністю рухів користувача під час навігації в межах віртуального середовища відеоігор [4]. Тобто подібні відеоігри несуть не тільки розважальний характер, але й розвиваючий.

Ця робота спрямована на адаптацію звичайних відеоігор під потреби сліпих, використовуючи тактильну і аудіо віддачу щоб передати оточення, а також надати бюджетну і розширювану платформу, для подальшого створення ігор на ній.

1.2. Проблематика

Незалежно від жанру та специфіки гри, обсяги інформації що користувач отримує через графічну складову занадто великі щоб їх можна було передати через аудіо і дотик. У 3D іграх ця задача постає навіть ще більш комплексною: позиціонування, довкілля, динамічні події, тощо. На жаль, спроба адаптувати більшість сучасних ігор до суто аудіо + тактильної віддачі безсумнівно приведе до суттєвих спрощень як ігрового процесу, так і інших складових.

Також, адаптація гравця до нових методів взаємодії з грою точно не буде миттєвою, а тому, для того щоб спростити занурення, не варто перевантажувати гравця інформацією. Не дивлячись на те, що в теорії можна використовувати безліч варіантів комбінації дотику і слуху, і необмежену кількість пристроїв для передачі почуття дотику (температура, вібрація, тощо) - є розумний ліміт, адже людина просто не зможе сприймати та вірно інтерпретувати таку кількість інформації одночасно. Через описані вище фактори, у практичній частині було накладено обмеження у 3 вібромодулі та звичайні навушники з стерео звуком.

1.3. Ехолокація

Основним методом донесення інформації користувачу є ехолокація. Ехолокація - це метод орієнтації в просторі, а також засіб виявлення розташування, розміру та структури об'єктів за допомогою відбитого звуку. Задля передачі найбільш повноцінної інформації про оточення гравця, потрібно використання комплексної системи ехолокації, яка має бути розроблена спеціально під конкретну гру, чи вузький жанр ігор. Разом з тим, деякими перешкодами у навчанні ехолокації є обсяги практики, яких вона вимагає, та когнітивне навантаження, яке вона створює. Незважаючи на те, що за підрахунками, від 18 до 32 відсотків сліпих людей, здатні використовувати ехолокацію на різних рівнях ефективності [5], досвід з реального світу безсумнівно буде відрізнятися від систем, створених під кожен з ігор індивідуально. Не дивлячись на це, попередні дослідження продемонстрували наявність прямої кореляції між використанням моторних навичок, та навичок просторової орієнтації у сліпих людей в віртуальному середовищі та у реальному житті [6].

Процес ехолокації складається з двох етапів: передача, тобто процес генерації звуку, та прийом – отримання відбитого звуку та інтерпретація результату. Однією з проблем симулювання ехолокації в відеогрі є швидкість самого процесу. Для того щоб отримати інформацію про те, що якийсь об'єкт знаходиться на великій відстані від місцерозташування гравця – потрібно лінійно більше часу. Це призводить або до штучного сповільнення темпу гри, або до підвищення складності обробки сигналів користувачем. З цієї причини, для донесення певних типів інформації про віртуальне оточення, було вирішено використовувати тактильну віддачу.

1.4. Ray casting

Метод що зазвичай використовується для реалізації концепту ехолокації у програмних рішеннях – рейкастинг [7] (англ. кидання променів). Рейкастинг – метод перетину променів з поверхнею для вирішення різних проблем у обчислювальній геометрії, та комп'ютерній графіці. Зазвичай, даний алгоритм для рендеру тривимірних сцен, проте його можна адаптувати під власні цілі. Наприклад, для визначення відстані від поточного об'єкта, до будь-якої іншої перешкоди на прямій лінії.

Розділ 2. Огляд використаних інструментів та технологій

2.1. Мікроконтролер

Для управління вібрomodулями необхідно організувати двосторонню комунікацію з ігровим рушієм у режимі реального часу, тому логічним буде використання мікроконтролера. Мікроконтролер – це спеціалізований мікроелектронний програмований прилад, що призначений для використання у керуючих пристроях, системах передачі даних та системах керування технологічними процесами, що включає мікропроцесор, оперативну та постійну пам'ять для збереження виконуваного коду програм і даних, порти вводу-виводу. На сьогодні існує велика кількість серійних лінійок як спеціалізованих контролерів, наприклад використовуваних в інженерії побутової техніки, так і контролерів-платформ, для аматорської розробки.

Для даної роботи було проаналізовано наступні платформи:

- Raspberry PI
- STM32
- MSP430
- Arduino

Основними критеріями вибору є швидкість розробки, ціна, потужність, та наявність готових бібліотек і інтерфейсів, так як самостійне створення драйверу для прямого обміну даних з ігровим рушієм по USB не є раціональним. Raspberry - найпотужніша з платформ, а також має зручні інструменти розробки, проте, варто зазначити що в даному випадку така потужність не є необхідною, а також цінова категорія платформи не задовольняє потреби проекту. У контролерів STM32 і MSP430 відсутні інтерфейси для комунікації з ігровим рушієм.

Через вищеперераховані критерії для проекту була обрана платформа Arduino Leonardo. Arduino Leonardo - це пристрій на базі мікроконтролера ATmega32U4. У його склад входить все необхідне для роботи з даним мікроконтролером: 20 цифрових входів / виходів (7 з яких можуть працювати в якості ШІМ-виходів), роз'єм мікро-USB, роз'єм живлення, роз'єм для внутрисхемного програмування ICSP і кнопка скидання. Для початку роботи з Leonardo досить просто подати живлення від AC / DC-адаптера або батареї, або підключити його до комп'ютера за допомогою USB-кабелю. Розробка для цього контролера ведеться на процедурній мові програмування C, з використанням спеціального фреймворку, та IDE для завантаження скомпільованого коду без зовнішніх програматорів. Однією з ключових переваг є розширюваність платформи (модульність), та її функціонал, а саме: ШІМ (Широтно-Імпульсна Модуляція), для контролю вихідного сигналу завдяки зміни ширини імпульсів, а не зміни вихідної напруги, а також наявність інтерфейсу UART, який дає змогу спілкування мікроконтролера з іншими пристроями, наприклад комп'ютером, використовуючи універсальний інтерфейс USB.

2.2. Ігровий рушій та комунікаційний інтерфейс

Для розробки відеоігор доцільним є використання вже існуючого ігрового рушія. Ігровий рушій – центральна програмна частина будь якої відеогри, відповідальна за більшу частину технічної сторони проекту, що дозволяє полегшити розробку гри шляхом уніфікації та систематизації її структури. Рушій бере на себе відповідальність за основний функціонал:

1. Рендеринг зображення
2. Фізичні симуляції
3. Керування пам'яттю та мережевим кодом
4. Файлову структуру

5. Звукову систему

6. Компіляцію та побудову проекту

Для даної роботи також є важливим можливість розширення функціоналу рушія, а саме імпортування, або побудова власних бібліотек та інтерфейсів комунікації. Додатково, перевагою буде багатofункціональна, та контрольована звукова система, для реалізації ехолокаційного методу.

Комунікація з мікроконтролером може відбуватися двома принципами: дротовим, та бездротовим. Перевагами бездротового методу є: портативність і комфорт, проте, варто зазначити, що даний метод створює й додаткові, критичні недоліки. По-перше, для цього необхідно щоб і пристрій на якому запущена гра мав бездротовий інтерфейс, наприклад Bluetooth, і у платформи був модуль бездротового зв'язку. По-друге, постає проблема інтерференції та шуму, тобто небажане порушення електричного сигналу. По-третє, обчислювальна потужність мікроконтролеру ардуіно – відносно мала, тому використання бездротового зв'язку займатиме важливу частину часу процесора, та оперативної пам'яті чіпу, що зменшить потенціальну та реальну швидкість обміну пакетами даних. З цієї причини було вирішено використовувати комунікацію через серійний USB порт.

У якості ігрового рушія був обраний Unity Engine, крос-платформний ігровий рушій, розроблений Unity Technologies, вперше анонсований та випущений у червні 2005 року. Він задовольняє усі вищеперераховані вимоги, а також має ще одну, неочевидну перевагу: мова програмування що використовується у даному рушії – C#, а це дає прямий доступ до Win32 API, який необхідний налагодження для комунікації по серійних портах (Serial/COM ports). Unity підтримує розробку як 2D так і 3D відеоігор, а також має вбудовану реалізацію фізичного рейкастингу, та потужний інструментарій для управління об'ємним стереозвуком.

У якості базового інтерфейсу для обміну даних між рушієм та мікроконтролером були розглянуті [8] три плагіни (розширення) для Unity: Uduino , Ardity, та wrhtml. Основним недоліком першого варіанту, Uduino, є відсутність підтримки багатопотоковості, що сповільнює та ускладнює двосторонній обмін пакетами. Проблемою wrhtml постає недостатній функціонал, та відсутність підтримки від розробника. З цих причин, на версіях Unity випущених після 2017 року, запропонована бібліотека не працює, та потребує значного доопрацювання.

Ardity – малий за обсягом плагін, розроблений dlswitches, який підтримує двосторонню комунікацію на основі багатопотоковості. Розробником були надані абстрактні класи для створення власних реалізацій серійних потоків (див. Додаток А). У практичній частині описано розширення функціоналу бібліотеки.

Розділ 3. Проектування та розробка системи

3.1. Апаратна складова

Апаратна частина складається з наступних компонентів:

- Arduino Leonardo (ATmega32U4)
- 3 x Плоских вібромоторів 8мм 3-5В
- 3 x RGB LED-Світлодіоди
- 2 x Резистори 320Ом
- Матеріали для прототипування (дроти, breadboard, USB)

З перерахованих вище складових була розроблена та зібрана електрична схема (див. Додаток Б). Живлення та контроль вібромодулів відбувається напряму через Arduino, через порти що підтримують Широтно-Імпульсну Модуляцію, для управління потужністю моторів. Світлодіоди під'єднані до тих самих цифрових виходів що і мотори, для відлагодження роботи, та наглядної демонстрації значень поточного сигналу. Резистор необхідний для безпечної роботи вібромоторів, та зниження шуму на лінії заземлення.

ШИМ дозволяє задавати значення у діапазоні від 0 до 255, що відповідає від 0% до 100% потужності роботи вібромодулів, при цьому не змінюючи напруги ланцюга, використовуючи принцип інерції. Перевагою такої схеми є її розширюваність і адаптація під необхідну гру, так як ардуіно здатен підтримувати до 7 подібних датчиків.

3.2. Прототипування

Для зручності використання прототипу, у Autodesk Fusion360 було створена модель кільця для комфортного розташування вібромодулів

(Додаток В). Fusion360 - це хмарний CAD / CAM-інструмент для розробки продуктів. Він поєднує в собі моделювання 3D форм, механічний дизайн та виготовлення в одному комплексному пакеті. Розроблена модель була надрукована в декількох екземплярах на 3D принтері, з матеріалу PLA+.

3.3. Основа проекту

Після аналізу варіантів, жанром гри було обрано автосимулятор у стилі картингу. У даному жанрі - з видом від першого або від третьої особи, в яких гравець бере участь в гоночному змаганні серед наземних транспортних засобів. У загальному випадку гоночної грою може бути комп'ютерна гра будь-якого жанру: від автосимуляторів високого ступеня реалізму до відносно простих аркадних гоночних ігор.

Причиною вибору жанру стали наступні критерії:

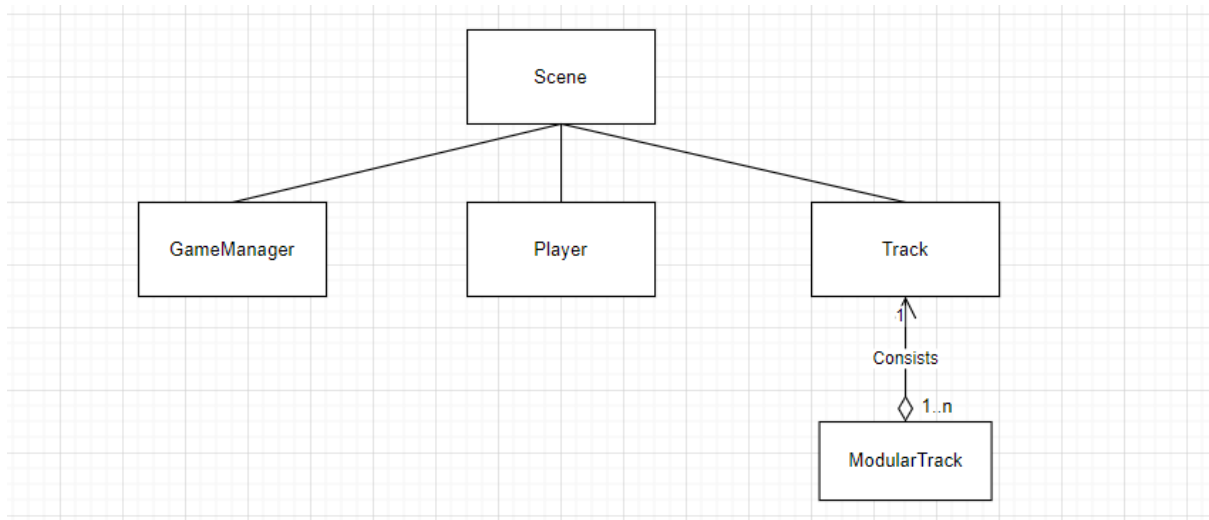
- Ефективність демонстрації розробленої системи
- Дії мають відбуватися в режимі реального часу
- Вплив відеогри на розвиток когнітивних і моторних навичок
- Простота сприйняття

На основі ігрового рушія Unity3D створено базисну відеогру Картинг. Частково, метою було створити гру що має мінімальну кількість спрощень, порівняно з повноцінним варіантом, адже за допомогою додаткового органу чуття – дотику, можливо передати значно більшу кількість інформації гравцю. Логічно, що графічна частина відеогри не є важливим фактором даній роботі, тому було використано готові 3D моделі гравця, автомобіля, та карти.

Карта є модульною, та побудованою з префабів. Префаби - це особливий тип компонентів, що дозволяє зберігати повністю налаштовані Ігрові Об'єкти у проекті для повторного використання.

Потім ці активи можна розподіляти між сценами або навіть іншими проектами без необхідності повторної конфігурації. Таке рішення дозволяє користувачам самим побудувати будь-яку карту з вже готових компонентів. Кожен з видів префабів модульного треку обов'язково має наступні компоненти:

- MeshCollider (Фізика)
- MeshRenderer (Рендер об'єкту)
- AudioSource (Ехолокація)



Контроль моделлю гравця відбувається за через скрипт `ArcadeKart.cs`. У цьому скрипті можна налаштувати параметри самого авто:

```

public struct Characteristics
{
    public float MaxSpeed;
    public float Acceleration;
    public float MaxSpeedReverse;
    public float AccelerationReverse;
    public float BrakingAcceleration;
}

```

```
}
```

Також, був створений скрипт для того щоб модель гравця завжди торкалася треку, і відбивалася від стін, тому що вищеперераховані ситуації можуть призвести до втрати управління, та дисонансу інформації у гравця який орієнтується лише на почуття дотику і звуку. Ключовими місцями коду даного модулю є використання рейкастингу для пошуку вектору нормалі з поверхнею від якої має відбитися об'єкт

```
if ( Physics.Raycast(originPoint, directionPoint, out RaycastHit hit, distance)
&& Time.time > resumeTime && !hasCollided && car.LocalSpeed() > 0 )
{
    if (Vector3.Dot(hit.normal, Vector3.up) > offset_angle)
    { return; }

    Vector3 incidentVec = hit.point - origin;
    Vector3 normal = hit.normal.normalized;
    angleOfReflection = incidentVec - 2 * Vector3.Dot(incidentVec, hitNormal) *
normal;
    angleOfReflection.y = 0;
    car.Rigidbody.velocity /= 2;
    car.Rigidbody.AddForce(angleOfReflection.normalized * BounceFactor,
ForceMode.Impulse);
}
```

3.5. Протокол обміну даних

Обмін даних між контролером та грою відбувається за допомогою інтерфейсу SerialThreadLines.cs, що наслідує абстрактний клас бібліотеки Ardity - AbstractThreadLines. Два основні методи для комунікації на стороні рушія це:

```

public string ReadSerialMessage()
{
    return (string)serialThread.ReadMessage();
}

```

та

```

public void SendSerialMessage(string message)
{
    serialThread.SendMessage(message);
}

```

Ці методи зчитують дані з черги, та передають дані у чергу відповідно. Основна проблема передачі даних – мікроконтролер може зчитувати інформацію лише по-байтово, тому на стороні мікроконтролера був написаний простий код для зчитування інформації по створеному протоколу. Комунікація двостороння – рушій відправляє інформацію зі значеннями вібромоторів, контролер – інформацію про успішне прийняття даних.

3.6. Метод ехолокації

Концепція ехолокації у реалізації проекту виконує наступні функції:

1. Інформації про подальший шлях до наступної контрольної точки
2. Інформація про фізичну колізію
3. Інформація про проходження контрольних точок
4. Інформація про завершення рівня

Сама мапа, складається з декількох типів об'єктів: модульні елементи треку, контрольні точки, які рекомендовано розміщати через кожні 5-10 модульних треків, та фініш/початок гонки. Мапа є циклічною.

Для того щоб гравець міг орієнтуватися на рівні, через рівну кількість модульних елементів дороги були розміщені контрольні точки. При натисканні на «Клавішу Ехолокації» скрипт TrackHandler.cs отримує поточний елемент дороги (на якій знаходиться гравець), та усі наступні елементи до найближчої контрольної точки. Далі, за допомогою компоненту 3D Audio Listener, що наявний у гравця, та компонентів Audio Source, які є на кожному елементі дороги, програватиметься певний звуковий ефект, у порядку розташування траси. Таким чином, гравець отримує стереозвук з кожного з наступних модулів, які йому потрібно проїхати, скорегований відносно відстані до модуля, та його розташування (Додаток Д).

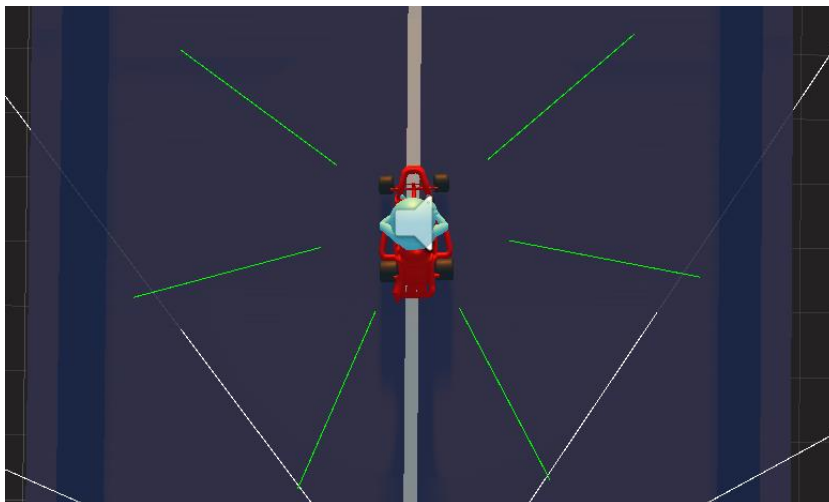
Інформація про колізію з стінкою - характерний звук KartBump. Також, при проходженні контрольних точок, та фінішу – програватимуться унікальні звукові ефекти. Звуки зроблені унікальним, для передачі конкретної інформації, і встановлення у гравця логічних зв'язків.

3.7. Використання тактильної інформації

У апаратній частині використані 3 електромотори. За допомогою вище описаного протоколу, мікроконтролер отримує інформацію про необхідні значення моторів, та передає сигнал на цифрові порти що підтримують PWM. Кожен з вібрomodулів відповідно надає гравцеві інформацію про:

1. Відстань від лівого краю карти
2. Відстань від правого краю карти
3. Швидкість

На рівні ігрового рушія, дана інформація вираховується як середнє зважений результати кидання трьох променів з праву частину машини, та трьох променів з лівої частини:



Далі, інформація про рейкастингу аналізується, обраховується та відправляється до мікроконтролеру. Відбувається динамічна зміна значень потужності електромоторів на стороні мікроконтролера, що дає гравцю додаткову інформацію про оточення, швидкість та місцезнаходження віртуального актора.

Висновки

У даній роботі було досліджено та побудовано апаратне-програмне рішення на основі мікроконтролеру Arduino, ігрового рушія Unity, та використанням зовнішніх бібліотек, з метою надати людям з вадами зору додаткові можливості у сфері відеоігор. Було розроблено відеогру у жанрі аркадного автосимулятора, у якому використовується ехолокаційний метод, та тактильну передачу інформації гравцеві за допомогою вищеперерахованого інструментарію.

Список використаних джерел

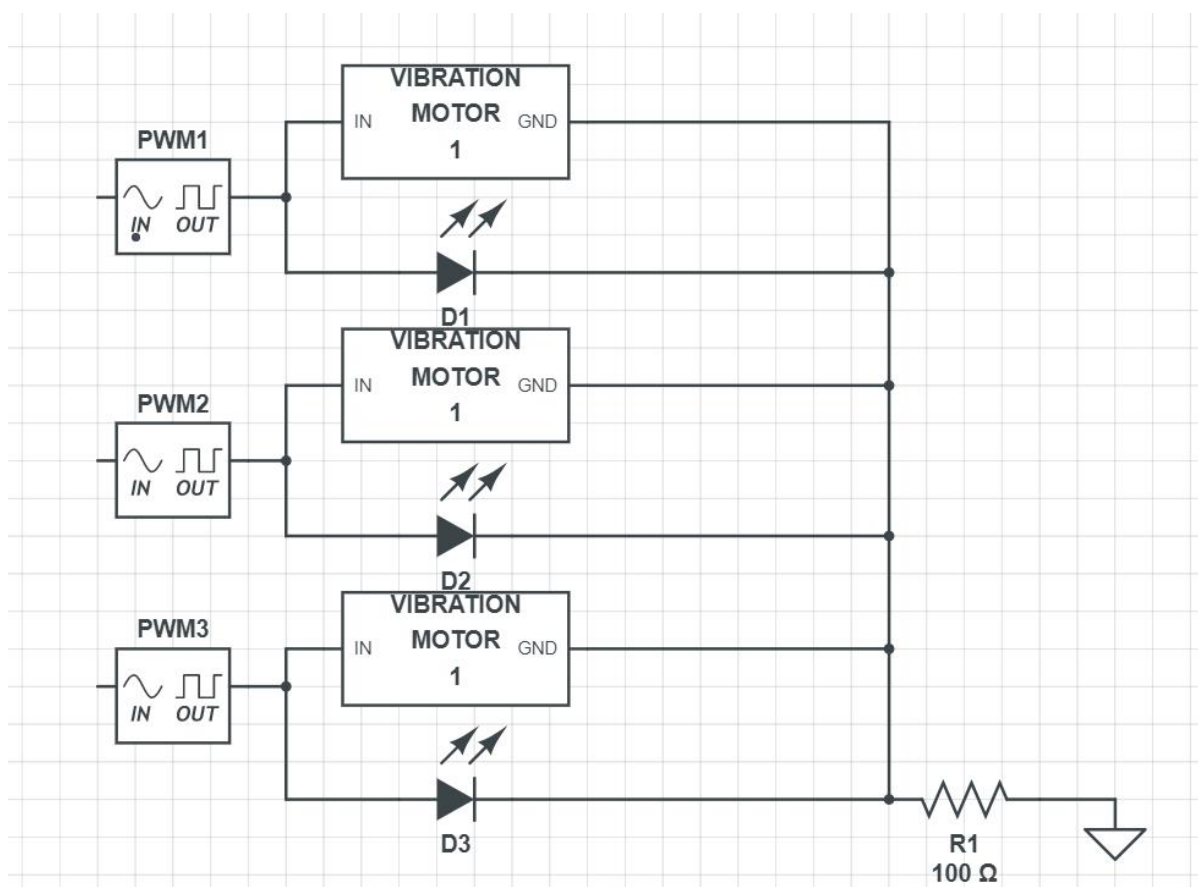
1. Озвучення UI у відеоіграх [Електронний ресурс] / Режим доступу до ресурсу: <https://www.minecraft.net/en-us/accessibility/>
2. Форум аудіоігор [Електронний ресурс] / Режим доступу до ресурсу: <https://forum.audiogames.net/>
3. Videogame controller for Tetraplegic [Електронний ресурс] / Режим доступу до ресурсу: <https://www.quadstick.com/>
4. Санчес, Дж., І Еспіноса, М. Аудіо-тактильні відеоігри для розвитку навичок навігації у сліпих учнів / Джейме Санчес, Матіас Еспіноза. // CARE. – 2011.
5. Талер Л. Ехолокація може мати реальні переваги для незрячих людей: аналіз даних опитування / Лоре Талер. // Frontiers In Psychology – 2013
6. Андраде Р. Echo-House: Дослідження віртуального середовища Використання ехолокації / Ронні Андраде, Стівен Бейкер, Дженні Вейкотт, Френк Ветере. // School of Computing and Information Systems – 2018
7. Свен Вуп. RPU: програмований блок обробки променів для трасування променів у реальному часі. / Свен Вуп, Йорг Шмітлер та Філіп Слусалек //ACM Trans. - (липень 2005), с.434–444.
8. Е. Кучера. "Зв'язок між єдністю 3D-двигуна та мікроконтролером arduino: віртуальний розумний будинок" / Е. Кучера, О. Гаффнер та Ш. Козак // Кібернетика та інформатика (К&І) - 2018

Додаток Б

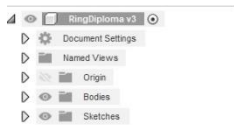
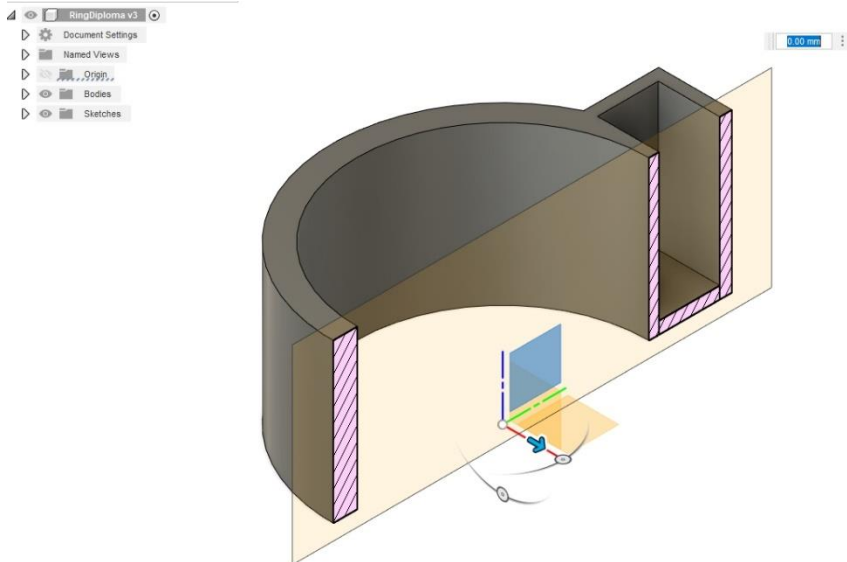
Електрична схема апаратної частини.

PWM – ШІМ

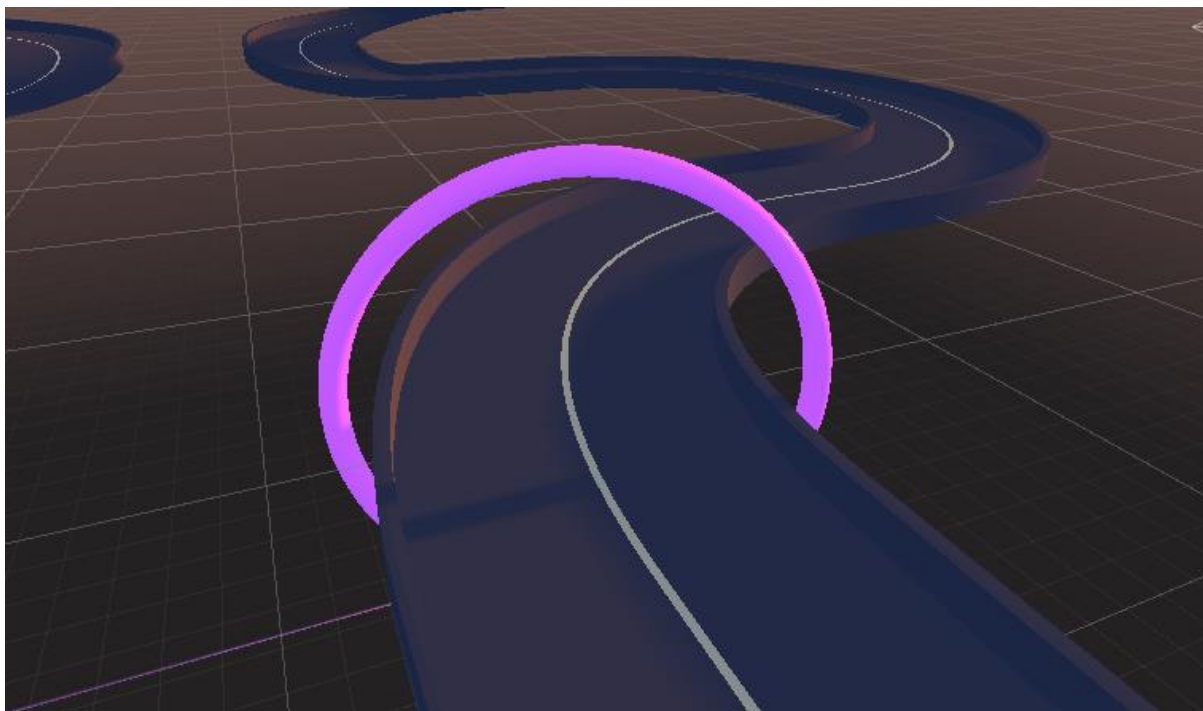
D1-D3 – світлодіоди



Додаток В

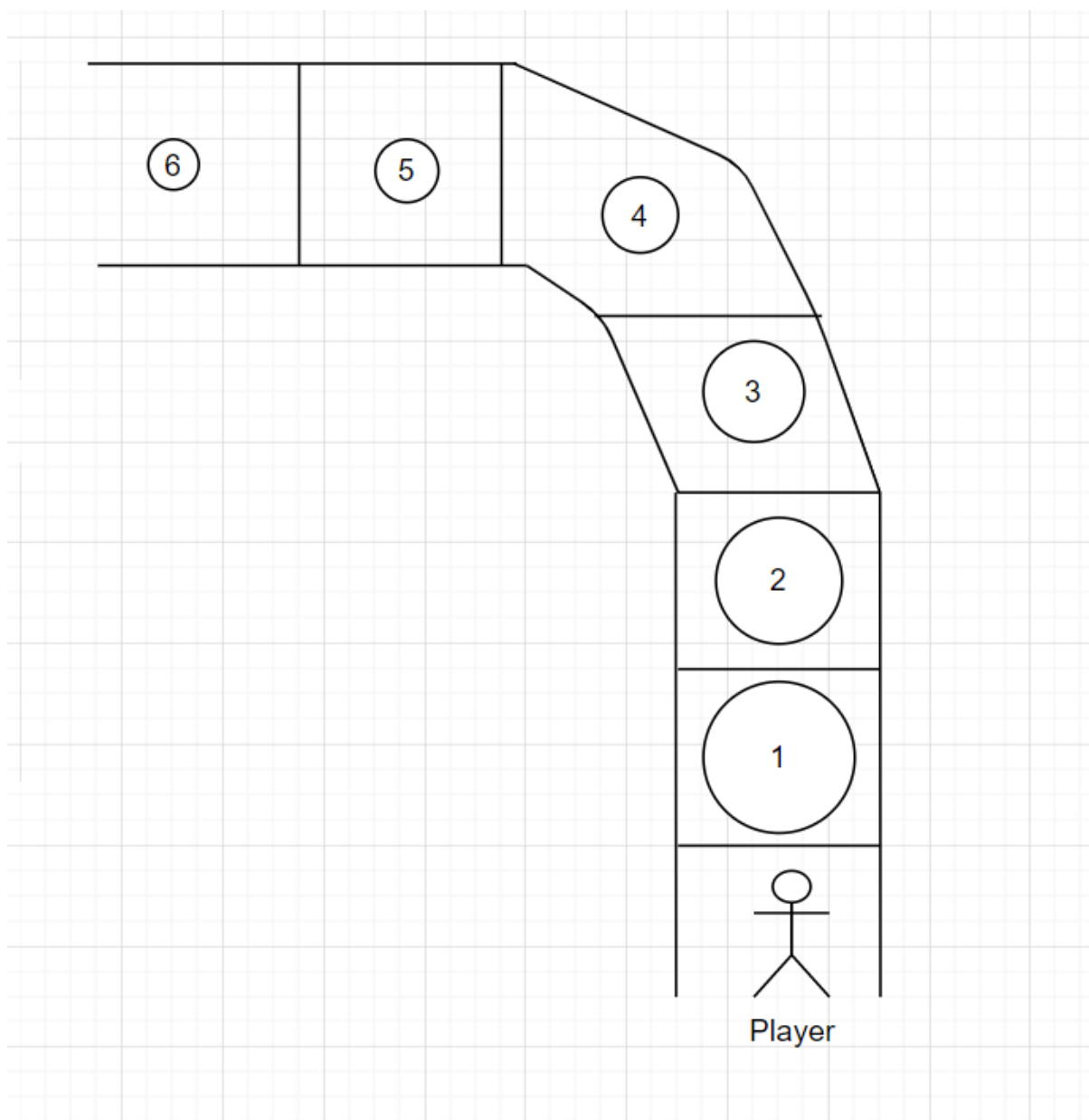


Додаток Г



Додаток Д

Радіус кола – загальна гучність ефекту.



Додаток Є

```
if (Serial.available() > 0) {  
  char *arr = new char[7];  
  int i = 0;  
  while (Serial.available() > 0) {  
    arr[i] = Serial.read();  
    ++i;  
  }  
  message = String(arr);  
  int indexOfDelimiter = message.indexOf(' ');  
  firstString = message.substring(0, indexOfDelimiter);  
  
  secondString = message.substring(indexOfDelimiter + 1);  
  updateMotorSpeed(firstString.toInt(),secondString.toInt());  
}
```