

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра автоматизації та інтелектуальних інформаційних технологій

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**  
на тему:

**Автоматизована система сортування об'єктів за QR-кодом**

Виконав: студент 2-го курсу, групи  
ЗАКІТ-22м  
спеціальності 151 – Автоматизація та  
комп'ютерно-інтегровані технології



**Андрій БАСАЙ**

Керівник: д.т.н., професор каф. КСУ



**Володимир ДУБОВОЙ**

« 4 » листопада 2023 р.

Опонент: к.т.н., доцент каф. КН



**Руслан БЕЛЗЕЦЬКИЙ**

« 4 » листопада 2023 р.

Допущено до захисту  
Зав. кафедри АІТ ВНТУ  
д.т.н., проф.

Олег БІСІКАЛО  
« 11 » листопада 2023 р.



ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра автоматизації та інтелектуальних інформаційних технологій  
Рівень вищої освіти другий (магістерський)  
Галузь знань – 15 Автоматизація та приладобудування  
Спеціальність – 151 Автоматизація та комп'ютерно-інтегровані технології  
Освітньо-професійна програма – Інформаційні системи і Інтернет речей



Допущено до захисту  
Зав. кафедри АІТ ВНТУ  
д.т.н., проф.  
Олег БІСКАЛО  
« 30 » вересня 2023 р.

**ЗАВДАННЯ**  
**НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ**  
студенту Басюю Андрію Руслановичу

1. Тема роботи Автоматизована система сортування об'єктів за QR-кодом  
керівник роботи д.т.н. проф. Дубовой В.М  
затверджені наказом вищого навчального закладу від "18" вересня 2023 року  
№247

2. Термін подання студентом роботи "12" грудня 2023 року

3. Вихідні дані до роботи:

1. Monica Burns. Deeper Learning With QR Codes and Augmented Reality: A Scannable Solution for Your Classroom / Corwin - First edition. – 2016 - с. 5-6, с. 17-18
2. Malburg, L., Seiger, R., Bergmann, R., Weber, B., 2020c. Using Physical Factory Simulation Models for Business Process Management Research, in: Del R'io Ortega, A., Leopold, H., Santoro, F.M. (Eds.), Business Process Management Workshops, Springer. pp. 95–107.
3. Object Detection for Smart Factory Processes by Machine Learning. Procedia Computer Science. / Lukas Malburg, Manfred-Peter, Seiger Ronny, Klein Patrick, Bergmann Ralph. – 2021р.

4. Зміст текстової частини:

- 1) Аналіз методів сортування об'єктів на конвеєрі

- 2) Розробка структури системи
- 3) Розробка програмного забезпечення
- 4) Тестування та дослідження програмного забезпечення

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Мета, предмет та практична цінність дослідження, головні завдання роботи  
приклад ідентифікації об'єктів на конвеєрних стрічках в промисловості, схеми роботи конвеєра, діаграма діяльності програмного забезпечення, алгоритм ідентифікації, розробка генератора QR-коду, тестування генератора QR-код  
розробка сканера QR-коду, тестування сканера QR-коду, висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали	Підпис, дата	
		Завдання видав	Виконання прийняв
1-4	Дубовой В.М.	21.09.2023	

7. Дата видачі завдання "09" жовтень 2023 року

#### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів Магістерської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз методів сортування	03.10.2023 р.	Вик.
2	Розробка структури системи	18.10.2023 р.	Вик.
3	Розробка програмного забезпечення	13.11.2023 р.	Вик.
4	Тестування програмного забезпечення	22.11.2023 р.	Вик.
5	Оформлення пояснювальної записки і графічного матеріалу	01.12.2023 р.	Вик.
6	Попередній захист	07.12.2023 р.	Вик.
7	Остаточний захист	16.12.2023 р.	Вик.

Студент

Андрій БАСАЇ

Керівник роботи

Володимир ДУБОВОЇ

## АНОТАЦІЯ

УДК 004.9:621

Басай А.Р. Автоматизована система сортування об'єктів за QR-кодом. Магістерська кваліфікаційна робота зі спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології, освітня програма – Інформаційні системи і інтернет речей. Вінниця ВНТУ, 2023. 113 с. На українській мові, 4 розділи, 40 рисунків, 1 таблиця, 64 джерела.

Магістерська кваліфікаційна робота присвячена дослідженню та розробці автоматизованої системи сортування об'єктів за QR-кодом. У ході роботи було проаналізовано методи сортування об'єктів на конвеєрі. Було розроблено структуру системи та самого програмного забезпечення. Проведено тестування та дослідження програмного забезпечення.

Основна мета цієї магістерської роботи зосереджена на всебічному дослідженні та вдосконаленні автоматизованої системи сортування об'єктів, що використовує технологію QR-коду. У процесі дослідження проводився широкий аналіз, оцінюючи сучасні технології та аналогічні системи, що використовуються в промислових умовах. Дослідження охоплювало дослідження та порівняльний аналіз існуючих алгоритмів сортування, поширених у цій галузі.

Ключовим результатом цього дослідження стала розробка програми в рамках рушія Unity, ретельно розробленої для генерування та сканування QR-кодів. Також додаток містить реалізацію алгоритму «QuickSort», що сприяє ефективному та точному сортуванню об'єктів на основі відсканованих QR-кодів.

На етапі перевірки та тестування ретельно перевірялася ефективність системи за різних умов, забезпечуючи її масштабованість, точність і надійність у сценаріях практичного розгортання.

Ключові слова: сортування, QR-код, Unity, конвеєр.

## ABSTRACT

Basai A.R. Automated system for sorting objects by QR code. Master's thesis on specialty 151 - Automation and computer-integrated technologies, educational program - Information systems and the Internet of Things. Vinnytsia VNTU, 2023. 113 c. In Ukrainian, 4 chapters, 40 figures, 1 table, 64 sources.

The master's thesis is devoted to the research and development of an automated system for sorting objects by QR code. In the course of the work, the methods of sorting objects on the conveyor were analyzed. The structure of the system and the software itself was developed. Software testing and research was conducted.

The main goal of this master's thesis is focused on comprehensive research and improvement of an automated object sorting system using QR code technology. In the process of research, a wide analysis was carried out, evaluating modern technologies and similar systems used in industrial conditions. The research involved a survey and benchmarking of existing sorting algorithms common in the field.

A key result of this research was the development of an application within the Unity engine, carefully designed to generate and scan QR codes. The application also includes the implementation of the "QuickSort" algorithm, which facilitates efficient and accurate sorting of objects based on scanned QR codes.

During the verification and testing phase, the system's performance was thoroughly tested under various conditions, ensuring its scalability, accuracy and reliability in practical deployment scenarios.

Keywords: sorting, QR code, Unity, pipeline.

**ВІДГУК**  
**керівника магістерської кваліфікаційної роботи**

студента Басая Андрія Руслановича  
(прізвище, ім'я, по батькові)  
на тему Автоматизована система сортування об'єктів за QR-кодом

Тема магістерської роботи актуальна, оскільки позначення об'єктів QR-кодом набуває все більшого розповсюдження, а задачі сортування зустрічаються на багатьох підприємствах логістики.

В роботі студентом визначено мету та поставлено задачі дослідження, здійснено аналіз методів сортування об'єктів на конвесрі, розглянуто використання QR-коду в системах автоматизованого сортування, існуючі автоматизовані системи у сфері сортування об'єктів, методи розпізнавання об'єктів, розроблено структуру системи та її апаратне забезпечення та програмне забезпечення, зокрема, генератора QR-коду та сканера QR-коду. Розроблено алгоритм сортування. Тестування показало коректність розробки.

Студент показав достатньо високий рівень підготовки, самостійність як у виборі теми, так і шляхів її реалізації, знання фахової літератури; підготовленості студента до прийняття сучасних рішень, умінь аналізувати необхідні літературні джерела, приймати правильні інженерні рішення, застосовувати сучасні системні та інформаційні технології.

В процесі роботи спостерігалися певні відхилення від графіку виконання.

Недоліки роботи: у розробленому програмному забезпеченні не передбачено інтерфейс з апаратною частиною з функцією перевірки результатів сортування.

Загалом магістерська кваліфікаційна робота відповідає спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології, заслуговує на оцінку «В», а її автор, Басай Андрій Русланович, заслуговує на присудження освітньої кваліфікації «Магістр» за спеціальністю «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інформаційні системи і Інтернет речей»

Керівник магістерської кваліфікаційної роботи

Професор каф. КСУ, д.т.н., професор



Володимир ДУБОВОЙ

**ВІДГУК**  
**опонента магістерської кваліфікаційної роботи**

студента Басая Андрія Руслановича  
(прізвище, ім'я, по батькові)  
на тему Автоматизована система сортування об'єктів за QR-кодом

Магістерська робота за темою «Автоматизована система сортування об'єктів за QR-кодом» є важливою для сучасного підприємства, все частіше розробляються системи на базі чи з використанням QR-коду, а методи сортування удосконалюються весь час.

Робота відображає високий рівень підготовки студента і передає його здатність працювати над складними інженерними завданнями.

У роботі студент визначив ціль та сформулював завдання для дослідження. Було проведено аналіз методів сортування предметів на конвеєрі, досліджено використання QR-кодів у системах автоматизованого сортування, розглянуто наявні автоматизовані системи у цій сфері, описано методи розпізнавання об'єктів. Крім того, була розроблена структура системи та створено необхідне апаратне та програмне забезпечення, включаючи генератор та сканер QR-кодів. Був розроблений алгоритм сортування, коректність якого була підтверджена під час тестування.

Однак, відсутність інтерфейсу між програмним забезпеченням та апаратною частиною, що відповідальна за перевірку результатів сортування, на жаль, не розкриває повний функціонал системи, що обмежує її роботу.

В цілому, магістерська кваліфікаційна робота відповідає спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології, заслуговує на оцінку «В», а її автор, Басай Андрій Русланович, заслуговує на присудження освітньої кваліфікації «Магістр» за спеціальністю «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інформаційні системи і Інтернет речей»

Опонент магістерської кваліфікаційної роботи

Доцент кафедри КН, к.т.н.



## ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ МЕТОДІВ СОРТУВАННЯ ОБ'ЄКТІВ НА КОНВЕЄРІ.....	7
1.1 Автоматизована системи сортування як невід'ємна частина сучасного підприємства.....	7
1.2 Використання QR-коду в системі автоматизованого сортування .....	13
1.3 Існуючі автоматизовані системи у сфері сортування об'єктів .....	17
1.4 Аналіз методів розпізнавання об'єктів .....	19
1.5 Аналіз методів ідентифікацій .....	21
Висновки до розділу .....	27
2 РОЗРОБКА СТРУКТУРИ СИСТЕМИ.....	28
2.1 Апаратне забезпечення.....	28
2.2 Загальна структура системи.....	34
2.3 Алгоритмічне забезпечення.....	39
2.4 Програмне забезпечення .....	46
Висновки до розділу .....	49
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	50
3.1 Програмне забезпечення “Unity” .....	50
3.2 Розробка генератора QR-коду .....	53
3.2.1 Розробка інтерфейсу генератора QR-коду.....	55
3.2.2 Створення програмного коду генератора QR-коду .....	57
3.3 Розробка сканера QR-коду.....	62
3.3.1 Розробка інтерфейсу сканера QR-коду .....	63
3.3.2 Розробка програмного коду сканера QR-коду .....	64
3.4 Розробка програмного коду сортувального алгоритму .....	72
Висновки до розділу .....	75
4 ТЕСТУВАННЯ ТА ДОСЛІДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ...	77
4.1 Тестування генератора QR-коду .....	77



4.2 Тестування сканера QR-коду .....	81
Висновки до розділу .....	84
ВИСНОВКИ.....	85
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	86
Додаток А – Технічне завдання (обов’язковий).....	93
Додаток Б – Лістинг програмної частини генератора QR-коду .....	96
Додаток В – Лістинг програмної частини сканера QR-коду .....	97
Додаток Г – Лістинг програмної частини сортування .....	99
Додаток Ґ – Ілюстративна частина (обов’язковий).....	100
Додаток Д – Протокол перевірки кваліфікаційної роботи (обов’язковий) .....	113

## ВСТУП

**Актуальність.** Сьогоднішні підприємства, які займаються сортуванням, наприклад посилок чи інших об'єктів, які можуть рухатися конвеєром на виробництві, все частіше звертаються до використання цифрових технологій для покращення управління фізичними одиницями та їх переміщеннями.

Автоматизована система сортування за QR-кодом може бути розглянута як частина індустрії "Internet of Things" (IoT)[1], оскільки вона використовує цифрові технології для покращення управління сортування. Система спрямована допомагати підприємствам знижувати витрати, підвищувати продуктивність і покращувати обслуговування клієнтів, що робить її потужним інструментом для багатьох галузей та бізнес-сегментів.

Автоматизована система сортування - це технологічне рішення, яке використовує QR-коди для ідентифікації та сортування об'єктів або товарів в промисловому або логістичному процесі. Така система може використовуватися в різних галузях, включаючи логістику, виробництво, розподіл товарів, управління складом та багато інших.

Система, яка використовує QR-коди в своїй основі, має високу швидкість і точність читання – що дозволяє ефективно сортувати об'єкти без помилок[2]. Через точність зростає швидкість оброблення інформації, тому інтеграція цієї технології підвищує ефективність в переміщенні об'єктів. Зі зростанням об'єму логістичних операцій і торговельних операцій системи сортування за QR-кодами дозволяють підприємствам оптимізувати процеси доставки та розподілу товарів.

Скасування рутинної роботи, всі задачі які потрібно виконувати на постійній основі, такі як сортування можна автоматизувати, персонал зможе займатися важливішою роботою у цей час.

Головною перевагою є можливості вдосконалення технології через інтеграцію з IoT і іншими технологіями, системи сортування можуть бути легко інтегровані з іншими сучасними технологіями, такими як Internet of Things (IoT), що підсилює їхню ефективність та можливості.

Сучасні технології, наприклад QR-коди є швидкими у читанні та можуть покращити точність сортування.

Систему легко адаптувати під різні об'єми роботи, автоматизовані системи здатні пристосовуватися під різні потоки роботи, також їх легко корегувати на інший об'єм виробництва без суттєвих часових та економічних витрат.

Матричні коди мають високий рівень безпеки та контролю: QR-коди можуть бути захищені від підробки, що робить системи надійними і безпечними.

Зростання електронної комерції: За останні роки споживачі все більше вибирають онлайн-шопінг, що робить логістику і сортування ще більш важливими для сучасних підприємств.

Загалом, автоматизовані системи сортування QR-кодами – є потужними сучасними інструментами для підприємств, що прагнуть оптимізувати свої логістичні та виробничі процеси, підвищити якість обслуговування та знизити витрати.

**Метою роботи** є технологічне оновлення системи сортування об'єктів за QR-кодом на базі рушія Unity, та оптимізація робочих процесів використовуючи алгоритм швидкого сортування. Дослідження спрямоване на вдосконалення алгоритмів обробки даних, що дозволить забезпечити швидке та точне визначення інформації з QR-кодів, а також виконання точної ідентифікації та класифікації об'єктів на основі цих даних.

**Завданням роботи** зосереджене на:

- по-перше на аналізі методів сортування об'єктів на конвеєрі;
- по-друге розробка структури системи;
- по-третє розробка програмного забезпечення;
- по-четверте тестування та дослідження програмного забезпечення;

**Предметом дослідження** є методи та засоби для зчитування та розпізнавання QR-кодів, та подальшої роботи з даними, а саме сортування.

**Новизна (Інноваційність) роботи** полягає в створенні новітньої автоматизованої системи сортування об'єктів за QR-кодом на базі рушія Unity, що є сучасним рішенням, через свій широкий інструментарій та швидкодію для

використання серед багатьох галузей індустрії та бізнесу. Головними перевагами є швидкість, в своїй структурі QR-коди мають вбудовані коректуючі коди, що дає можливість містити в одному коді багато інформації.

**Методи дослідження:** для вирішення поточної задачі у роботі було використані методи та засоби вимірювального контролю, планування та розробка системи на базі багатоплатформового рушія Unity. Проведення експериментів у ході досліджень, моделювання, вимірювання похибок та проведення технічного контролю.

**Практична цінність** полягає в удосконаленні програмного забезпечення системи сортування об'єктів за QR-кодами використовуючи сучасний рушій Unity.

**Апробація.** Представлені в роботі результати апробовані в результаті участі в конференції Всеукраїнська науково-технічна конференція професорсько-викладацького складу, науковців, аспірантів та студентів підрозділів університету з участю працівників підприємств «LIII Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024)».

**Публікація:** Басай А.Р. Дубовой В.М. «Автоматизована система сортування об'єктів за QR-кодом», «LIII Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024)».

## 1 АНАЛІЗ МЕТОДІВ СОРТУВАННЯ ОБ'ЄКТІВ НА КОНВЕЄРІ

1.1 Автоматизована системи сортування як невід'ємна частина сучасного підприємства

Автоматизація – це заміна ручної праці на механічну, і вона економічно доцільна там, де потрібним максимальний дохід в довгостроковій перспективі безвідмовної роботи [5]. Автоматизація має недоліки, найважливіший – це негнучкість. Як правило, вона добре виконує поставлені конкретні задачі, для яких система була розроблена, але його адаптація під інші проекти може бути дорогою для підприємства [3].

Фіксовані автоматизовані системи, також відомі як жорстка автоматизація [14], призначені для повторного виконання одного набору завдань. Фіксовані автоматизовані системи [12] зазвичай використовуються для дискретного масового виробництва або систем безперервного потоку. Прикладом фіксованої автоматизованої системи є автоматизований конвеєр у автомобільній промисловості [13], який переміщує об'єкти з мінімальним зусиллям, підвищуючи ефективність.

Програмовані автоматизовані системи керуються командами, які надсилаються комп'ютерною програмою. Через це автоматизовані процеси можуть варіюватися в залежності від інструкцій, надісланих комп'ютером за допомогою коду розробника [15]. Програмована автоматизація широко використовується в умовах, де виробляються схожі вироби за допомогою однакових автоматизованих кроків і інструментів [4]. Наприклад, виробництво паперу та сталевих прокатних станів використовує однакові кроки для створення багатьох різних типів продуктів.

Гнучка автоматизація, також відома як м'яка автоматизація [14], зазвичай використовується в пакетних процесах з різноманітними продуктами. Кожному обладнанню надаються інструкції для комп'ютера, яким керує людина, тому

зміна коду може бути передана комп'ютеру, що дозволяє більш гнучке виробництво. Основною перевагою гнучкої автоматизації [11] є те, що зміна продукції відбувається швидко і автоматично, оскільки вона передається системою керування, що усуває додатковий час, необхідний для налаштування обладнання між партіями. Галузі, які використовують гнучку автоматизацію, включають текстильне виробництво, переробку харчових продуктів та виробництво фарб.

Інтегровані автоматизовані системи [12] передбачають повну автоматизацію виробничих підприємств з мінімальною участю людини. Комп'ютери можуть використовуватися для проектування деталей [15], перевірки готових проектів та виготовлення нових деталей. Роботизована система сортування зображена на рис. 1.1. Інтегрована автоматизація може бути застосована в процесі безперервного виробництва та пакетного виробництва [57].



Рисунок 1.1 – Роботизована система автоматизація [57]

Автоматизація, природно, зосереджена на зменшенні людського впливу, дотепер вона приймає форму модулів, призначених для роботи в якості системи пошуку та зберігання об'єктів.

Ця частина зосереджена на трьох питаннях автоматизації [10], перший і найнижчий рівень – контроль, зокрема, за допомогою яких алгоритмів слід отримувати об'єкт. Відповідь на це питання забезпечує основний робочий протокол автоматизації [9].

Останнім і найголовнішим масштабним питанням є оцінка пропускну́ї здатності [40] – з якою швидкістю можна отримати потік об'єктів?

Сортування об'єктів - це важлива частина логістичних та виробничих процесів у різних галузях [8]. Розвиток цієї області став можливим завдяки поєднанню ряду технологій, методів та інновацій .

Історичний розвиток методів сортування об'єктів ілюструє траєкторію від ручного сортування в стародавні часи до інтеграції передових технологій у сучасні системи [16]. Спочатку переважало ручне сортування, яке спиралося на людську класифікацію та впорядкування об'єктів на основі їхніх властивих характеристик або цільового використання. Ця практика зберігалася до появи машин, що стало ознаменуванням значного переходу до механізованих методів сортування. Протягом минулих століть еволюція машин і конвеєрів проклала шлях до автоматизації аспектів процесів сортування [15] у сферах виробництва та логістики, що, як наслідок, сприяло підвищенню продуктивності та зменшенню витрат на робочу силу [5, 6].

Ключовий момент виник із впровадженням штрих-кодів [7]. Штрих-коди революціонізували ландшафт сортування, уможлививши автоматизовану ідентифікацію та категоризацію товарів через стандартизовану структуру, підвищивши читабельність і ефективність обробки. Одночасно поява технології радіочастотної ідентифікації (RFID) [7] набула популярності як інструментального інструменту для відстеження та сортування об'єктів, використовуючи мітки RFID для полегшення відстеження об'єктів у реальному часі та на великій відстані.

Зараз поєднання QR-кодів і передових технологій стало повсюдним у сфері сортування та відстеження об'єктів. Інтеграція QR-кодів ознаменувала вирішальний крок, пропонуючи широкі можливості зберігання інформації, адаптовані до різноманітних програм [2]. Крім того, поширення технологій Інтернету речей (IoT) [1] революціонізувало методології сортування завдяки інтеграції датчиків в об'єкти, дозволяючи віддалений моніторинг і контроль через підключення до мережі.

Сучасні системи сортування були підкріплені інтеграцією машинного навчання та можливостей комп'ютерного зору [20]. Ці системи вміло використовують ці технології, щоб розпізнавати та класифікувати об'єкти, таким чином спрощуючи складні завдання сортування [9]. Крім того, розвиток робототехніки та автоматизованих систем суттєво підвищив ефективність сортування в різних галузях промисловості, сприяючи швидкій і точній категоризації об'єктів [4].

Крім того, використання хмарних служб і аналітики стало ключовим аспектом оптимізації сортування [3]. Використання цих інструментів дозволяє збирати, аналізувати та подальшу оптимізацію сортування даних для підвищення загальної ефективності. Еволюційна траєкторія сортування об'єктів не припиняється, і передові технології, такі як штучний інтелект, Інтернет речей і робототехніка [1], впроваджують нові можливості, що ще більше покращує автоматизацію в цій галузі.

Прикладом автоматизованих систем сортування є автоматизована конвеєрна система сортування. Ця система функціонує шляхом автоматичного сортування об'єктів, що транспортуються вздовж конвеєрної стрічки [6], і знаходить застосування в різних галузях, включаючи логістику, виробництво та складування. Основна передумова передбачає проходження об'єктів через сортувальні станції, оснащені датчиками, роботами чи іншими спеціалізованими пристроями для ефективного та безперебійного виконання процесу сортування [5]. Розгалужена система сортування – рис. 1.2.





Рисунок 1.2 – Розгалужена система автоматизованого складського комплексу [6]

Основні етапи автоматизованої системи сортування на конвеєрі можуть включати [18]:

**Зчитування інформації:** Система може використовувати різні способи зчитування інформації про об'єкти, такі як штрих-коди, RFID-мітки або візуальне розпізнавання.

**Класифікація:** Зчитана інформація про об'єкти допомагає системі визначити, до якої категорії вони належать. Наприклад, товари можуть бути класифіковані за розміром, кольором або типом.

**Розподіл:** Після класифікації система визначає, куди потрібно направити кожен об'єкт. Це може бути здійснено за допомогою роботизованих рук, конвеєрних відгалужень або інших механізмів.

**Переміщення:** Після розподілу система переносить об'єкти до відповідного місця, використовуючи рухомі деталі конвеєра або роботів.

**Перевірка:** Деякі системи сортування можуть включати етап перевірки якості або ідентифікації об'єктів перед їх відправкою або подальшою обробкою.

Переваги автоматизованої системи сортування на конвеєрі включають підвищену швидкість та точність сортування, зменшення людського втручання,

покращення ефективності виробництва та зменшення витрат. Вона також дозволяє забезпечити стабільність та надійність процесу сортування.

Сьогодні популярні два види конвеєрів – роликівий та стрічковий [5, 6]. На вибір між роликівим та стрічковим конвеєром впливає галузь застосування, а також потреби складу та їх завдання.

Роликівий конвеєр підходить для транспортування великих та важких товарів – палет, ящиків. Ролики дозволяють зменшити опір руху, що забезпечує зручне переміщення великих товарів. Стрічковий конвеєр підходить для транспортування легких та невеликих товарів – коробок та полібегів [6].

Стрічковий конвеєр є одним з найбільш універсальних транспортерів, який застосовується в різних галузях складської логістики, включаючи виробництво, харчову промисловість, фармацевтику, кур'єрські служби. Використання стрічкового конвеєра може відрізнитися залежно від галузі та вимог до транспортування вантажу.

Наприклад, у промисловій галузі стрічковий конвеєр [49] використовується для переміщення вантажів у зонах виробництва, транспортування матеріалів з одного робочого місця на інше, а також для переміщення товарів на лініях комплектації та відправлення. В автомобільній промисловості він може використовуватися для переміщення автомобільних деталей під час виробництва та збирання автомобілів. У харчовій промисловості стрічковий конвеєр може бути використаний для транспортування продуктів [50] від одного етапу виробництва до іншого, а також для упаковки та сортування продуктів. У фармацевтичній промисловості його можна використовувати для транспортування лікарських засобів та матеріалів у зоні виробництва.

Стрічковий конвеєр може бути виготовлений таким чином, щоб він був стійким до різних кліматичних умов [51], таких як низькі температури, висока вологість, пил та забруднення. Для цього використовуються матеріали, які легко очищаються. Це особливо важливо у харчовій та фармацевтичній промисловості, де вимоги до гігієни та безпеки є особливо суворими.

Отже, стрічковий конвеєр є універсальним транспортером, який знаходить застосування в різних галузях складської логістики [52]. Він може бути використаний для переміщення вантажів у зонах виробництва, транспортування матеріалів, комплектації та відправлення товарів. Крім того, він може бути стійким до різних кліматичних умов і вимог до гігієни та безпеки.

## 1.2 Використання QR-коду в системі автоматизованого сортування

**QR-коди** (Quick Response Code) [2] відіграють ключову роль у вирішенні цих завдань. Вони представляють собою багатофункціональні двовимірні штрих-коди, що здатні зберігати велику кількість інформації та забезпечувати швидкий доступ до неї. QR-коди можуть бути технологією для ідентифікації та сортування об'єктів у різних галузях [21].

QR-коди мають кілька ключових характеристик:

**Двовимірність:** QR-коди складаються з чорних квадратів, що розташовані на білому фоні. Ця двовимірність дозволяє зберігати значно більше інформації порівняно з одновимірними штрих-кодами [55].

**Великий обсяг зберігання інформації:** QR-коди можуть зберігати числа, букви, символи та інші дані. Розмір коду визначається кількістю інформації, яку необхідно закодувати.

**Швидкість зчитування:** QR-коди можуть бути зчитані швидко та точно за допомогою спеціальних сканерів або камер смартфонів.

**Функція корекції помилок:** QR-коди містять в собі механізми для виявлення та виправлення помилок. Це дозволяє відновлювати інформацію навіть при невеликих пошкодженнях коду [22].

**Різноманітність застосувань:** QR-коди застосовуються у різних галузях, включаючи логістику, маркетинг, медицину, туризм, торгівлю тощо.

**Компоненти QR-коду** включають:

-**Зона зчитування (Finder Pattern):** Це зліва вгорі та зліва внизу знаки, які дозволяють сканеру розпізнати QR-код і визначити його розмір та орієнтацію.

-**Зона даних (Data Area):** Це головна частина QR-коду, де зберігається інформація. Вона складається з чорних та білих квадратів.

-**Зона корекції помилок (Error Correction Area):** QR-коди містять додаткові дані для виявлення та виправлення помилок, що можуть виникнути при зчитуванні.

-**Зона контролю (Quiet Zone):** Ця зона складається з білих смуг, що оточують QR-код і дозволяють сканеру правильно виділити код.

QR-коди можуть зберігати різні види інформації, такі як текст, URL-адреси, контактні дані, географічні координати, Wi-Fi мережі та багато іншого. Ця універсальність робить їх ідеальним інструментом для ідентифікації та сортування об'єктів у різних галузях [55].

Сучасні виробничі та логістичні виклики вимагають постійного удосконалення та оптимізації процесів сортування об'єктів, щоб відповідати зростаючим обсягам та різноманітності завдань. У цьому контексті використання QR-кодів для ідентифікації та сортування стає перспективною технологією, яка може вирішити численні завдання у різних галузях [37].

Розглядається ця технологія в контексті її потенціалу для вирішення завдань оптимізації логістичних процесів та підвищення продуктивності виробничих систем.

Оптимізація та удосконалення процесів сортування об'єктів [16] стають ключовими завданнями для компаній, що прагнуть розвиватися. У цьому контексті виникає потреба в нових інструментах та технологіях, які здатні впоратися з великими обсягами та різноманітністю завдань [15].

На сьогоднішній день існує багато різних систем сортування і відстеження об'єктів у різних галузях індустрії, і кожна з них має свої переваги та недоліки. Порівняємо систему автоматизованого сортування за QR-кодами з іншими існуючими рішеннями:

Системи сортування на основі RFID: RFID (Radio-Frequency Identification) - це технологія, яка використовує радіочастотні мітки для ідентифікації та відстеження об'єктів [7]. Переваги систем RFID включають велику дальність читання, можливість відстеження в реальному часі та здатність працювати в умовах, де QR-коди можуть бути неефективними (наприклад, при поганому освітленні або в середині коробок). Однак RFID-системи зазвичай коштують дорожче в імplementації і можуть вимагати більшого обладнання.

Системи машинного навчання та комп'ютерного бачення - іншим інноваційним підходом є використання систем машинного навчання та комп'ютерного бачення [13] для сортування та відстеження об'єктів без необхідності QR-кодів або RFID-міток. Ці системи можуть бути здатними розпізнавати об'єкти на основі їхнього зовнішнього вигляду та форми. Однак вони можуть вимагати великих обсягів даних для тренування та можуть бути складними в імplementації.

Мануальне сортування: В деяких випадках, залежно від об'ємів та виду об'єктів, мануальне сортування залишається дешевшою та більш доступною опцією, але воно може бути повільним і піддається людським помилкам [16].

Системи геолокації та GPS: В деяких випадках, особливо в логістиці та транспорті, системи геолокації та GPS можуть використовуватися для відстеження та маршрутизації об'єктів [14]. Вони надають велику точність у визначенні місця розташування, але не завжди є практичними для всіх типів об'єктів.

Системи відстеження за допомогою баркодів: Баркоди є іншим поширеним методом відстеження та ідентифікації об'єктів [22]. Вони схожі на QR-коди, але можуть мати обмежену кількість інформації, яку вони можуть містити, і вимагають точного позиціонування для зчитування.

Вибір системи залежить від вашого конкретного бізнесу, його потреб і бюджету. Система автоматизованого сортування за QR-кодами може бути зручною та витратно-ефективною альтернативою, особливо для тих галузей, де QR-коди вже використовуються чи легко інтегруються.

Систему автоматизованого сортування за QR-кодами можна легко інтегрувати в багато різних галузей та видів бізнесу [19]. Ось декілька галузей, де така система може бути особливо корисною та легко інтегрованою:

Логістика та поставки: Система може бути використана для сортування та відстеження товарів під час їхньої транспортування та доставки. Вона допоможе визначити місцезнаходження товарів, відстежити їхній рух, ідентифікувати їх та оптимізувати логістичні процеси. [21]

Виробництво і виробнича лінія: У виробничих умовах система може бути використана для відстеження виробничих процесів, контролю якості та сортування виробів на різних стадіях виробництва.

Медицина і охорона здоров'я: QR-коди можуть бути використані для ідентифікації пацієнтів, медичного обладнання та медикаментів. Система може допомогти відстежувати потік ресурсів та забезпечити безпеку пацієнтів.

Роздрібна торгівля: QR-коди використовуються в маркетингових акціях, на етикетках товарів та в мобільних додатках для зручного визначення цін і іншої інформації про товари.

Бібліотеки і освіта: Система може використовуватися для автоматизації видачі та повернення книг в бібліотеках або для контролю студентського обліку у навчальних закладах.

Спорт та фітнес: QR-коди можуть бути використані для реєстрації та відстеження учасників спортивних заходів або клієнтів фітнес-центрів.

Туризм і гостинниці: QR-коди можуть надавати туристам доступ до інформації про атракції, ресторани та готелі, а також допомагати в бронюванні та вході.

Виробництво харчових товарів: У галузі харчової промисловості QR-коди можуть бути використані для відстеження походження та якості продуктів.

Експедиційні служби і кур'єрські послуги: Система може сприяти точному відстеженню пакетів та вантажів під час їхньої доставки.

Музеї та культурні заклади: QR-коди можуть надавати відвідувачам доступ до додаткової інформації про експонати та виставки.

Це лише декілька прикладів галузей, де система сортування за QR-кодами може бути інтегрована [57]. Важливо враховувати, що ефективність і доцільність інтеграції будуть залежати від конкретних потреб бізнесу та характеру операцій.

### 1.3 Існуючі автоматизовані системи у сфері сортування об'єктів

Зараз на виробництвах велика увага приділяється системам ідентифікації та управління процесами, які в основному ґрунтуються на дії датчиків, що реєструють дискретні значення. Ці датчики можуть відображати внутрішній стан окремих виробничих машин, реагувати на світлові бар'єри, активувати перемикачі або використовувати смарт-теги. Вони використовуються для постійного моніторингу та контролю за ходом виробничих процесів всередині лінії виробництва [21].

Але не всі процеси підпадають під радіус виявлення датчиками, через недостатнє покриття окремих місць на виробництві. В такому випадку до систем з існуючими процесами ідентифікації додають методи ідентифікації об'єктів на основі відео з метою забезпечення надійнішого моніторингу процесів, на основі кількох взаємопов'язаних модулів.

Сортування, яке виконується безпосередньо працею людей супроводжується високими матеріальними витратами та помилками, які виникають під час виконання процесів, зумовлені людським фактором.

Сортування об'єктів на основі QR-коду відбувається за рахунок визначеного спеціального параметра, наданого виробником. Це дає можливість зробити конвеєр більш гнучким та зменшити похибки.

Системи для сортування та ідентифікації об'єктів використовують не тільки в промисловості, а також в сільському господарстві та харчових підприємствах, де сортування та контроль відіграють найважливішу роль.

Виявлення та усунення (сортування) на основі QR-коду допоможе перетворити процес сортування, який зазвичай відбувається за рахунок праці

співробітників, на ефективний та автоматизований процес, збільшуючи при цьому рівень точності виявлення браку [35].

Метод автоматизованого зчитування об'єктів через камеру, також може використовуватись для надсилання сповіщень, автоматичного зупинення механізму чи надання координат розташування об'єкта для подальшого втручання системою, наприклад, для видалення непотрібних об'єктів. [13, 34].



Рисунок 1.3 – Процес виявлення небажаного предмету на сортувальній лінії [8]

Інтеграція алгоритмів зчитування даних через камеру [36], зокрема в поєднанні з технологією візуального контролю [42], розширює можливості розпізнавання в комп'ютеризованих системах [39]. Це об'єднання полегшує ідентифікацію та класифікацію компонентів, відхилень і символічних уявлень, імітуючи проникливі здібності візуального огляду людини [13]. Це злиття технологій має велике значення в різних секторах, зокрема в промислових і виробничих процесах, де точний контроль і виявлення є першорядними.

Ефективність зчитування даних через камеру полягає в його здатності просіювати величезні обсяги даних, виявляючи нюанси, які можуть вислизати від традиційних алгоритмів [20]. Завдяки ітераційному вдосконаленню параметрів ці системи досягають підвищеного рівня точності в розпізнаванні та класифікації різноманітних візуальних елементів.



Основний принцип передбачає навчання цих мереж за допомогою наборів даних, що дозволяє машині розрізняти різні функції та класифікувати їх із помітним ступенем точності [17].

Його інтеграція з технологіями візуального контролю демонструє надзвичайний потенціал, виходячи за межі традиційних парадигм, щоб революціонізувати розпізнавання образів, виявлення аномалій та символічну інтерпретацію в комп'ютеризованих системах у різноманітних областях [17].

#### 1.4 Аналіз методів розпізнавання об'єктів

Вивчення розпізнавання об'єктів [13] є центральною сферою в технічній науці, зосереджуючись на формулюванні принципів і побудові систем, спеціально призначених для ідентифікації та категоризації об'єктів на основі заздалегідь визначених критеріїв [15]. Об'єкти, що охоплюють різноманітні матеріальні об'єкти, продукти, явища, сигнали та технічні процеси, утворюють центр аналізу в цій дисципліні.

Основна мета полягає в імітації можливостей людського зору за допомогою програмних механізмів, намагаючись відтворити та перевершити проникливі здібності людського ока [42]. Центральним у цьому прагненні є здатність диференціювати та призначати класи об'єктам, зображеним у візуальних представленнях, завдання, яке підкреслює суть розпізнавання об'єктів [16].

За своєю суттю, отримання інформації із зображень включає складний процес обробки та аналізу даних [37]. Цьому виділенню сприяє модель, яка об'єднує принципи, отримані з геометрії, фізики, статистики та теорії навчання, утворюючи багатогранну структуру для інтерпретації та розуміння візуальних даних.

Геометрія вносить фундаментальні принципи, допомагаючи просторовому розумінню конфігурацій об'єктів на зображеннях [38]. У гру вступають

концепції, засновані на фізиці, які пояснюють взаємодію світла, тіней і перспективи у формуванні візуальних уявлень. Статистичні методології пропонують основу для імовірнісного аналізу, що дозволяє інтерпретувати закономірності та варіації в візуальних даних.

Теорія навчання, важливий компонент, використовує алгоритми та моделі для наповнення систем здатністю навчатися та покращувати точність їх розпізнавання з часом [19]. Ця парадигма навчання передбачає доступ до позначених наборів даних, що дозволяє системам удосконалювати свої здібності розрізнення через ітераційні процеси навчання.

Еволюція методів розпізнавання об'єктів [38] ґрунтується на постійному вдосконаленні алгоритмів, обчислювальних можливостях і розширенні анотованих наборів даних. Проблеми залишаються, зокрема мінливість зовнішнього вигляду об'єктів, фактори навколишнього середовища, що впливають на якість зображення, і пошук надійності проти непередбачених сценаріїв.

По суті, методології розпізнавання об'єктів представляють собою злиття різноманітних наукових дисциплін, які об'єднуються для відтворення та покращення можливостей людського зору за допомогою програмно-керованих систем [43]. Прагнення до вдосконалення цієї методології продовжує розширювати межі технологічних інновацій, впливаючи на різні сфери, від промислової автоматизації до медичної діагностики тощо [13].

На рис. 1.4 представлено приклад використання ідентифікації об'єктів на конвеєрі в промисловості.

Основна мета процесу ідентифікації полягає в побудові оптимальної моделі [41], реалізованої на основі ретельного спостереження та аналізу вхідних і вихідних змінних у системі. Ідентифікація, яка бере свій початок у сучасній теорії управління, значною мірою залежить від використання сучасних обчислювальних систем і методів для її вирішення.

Ідентифікація складних і багатогранних об'єктів у промисловому виробництві стала головною проблемою при побудові систем управління для цих

суб'єктів. Вирішення труднощів ідентифікації є фундаментальною передумовою, яка служить метою для цілеспрямованого проектування систем управління та створення адаптивних структур, здатних працювати в умовах, що характеризуються неповнотою інформації та розмаїттям об'єктів [16].

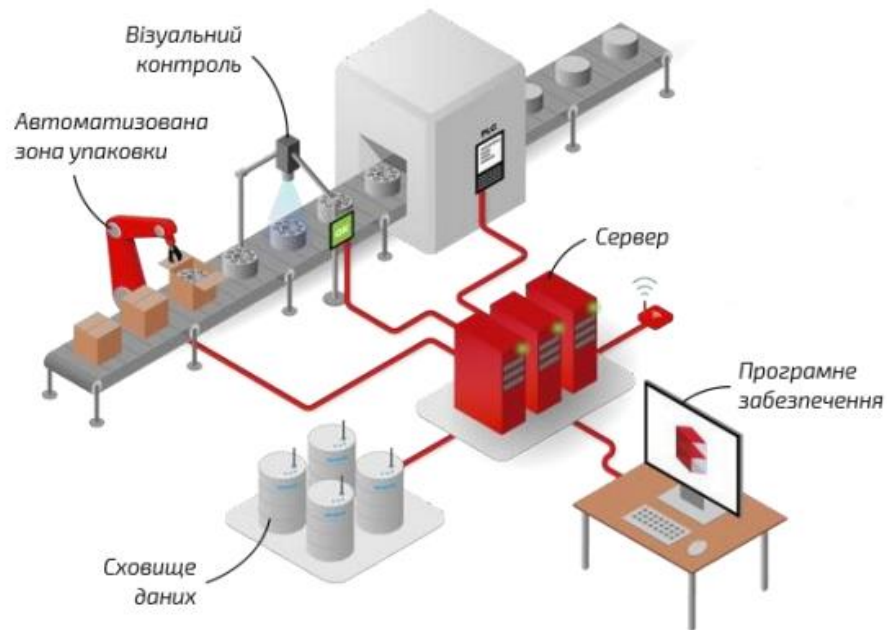


Рисунок 1.4 – Приклад ідентифікації об'єктів на конвеєрних стрічках в промисловості [12]

### 1.5 Аналіз методів ідентифікацій

Ідентифікація автоматизованих систем [59], яка вважається надзвичайно важливим аспектом, бере на себе роль першорядної початкової фази в операційній ієрархії. Успішне розв'язання труднощів ідентифікації стає не просто необхідністю, а критично важливим фактором для наступних етапів розвитку та ефективної оркестровки механізмів контролю та адаптації.

По суті, суть ідентифікації в сфері автоматизації полягає в її внутрішній здатності розшифрувати основну динаміку та моделі поведінки складних систем [60]. Шляхом асиміляції та аналізу взаємозв'язків «вхід-вихід»

ідентифікація намагається розкрити приховану структуру та функціональні аспекти системи, сприяючи створенню моделей, які інкапсулюють поведінку системи.

Цей процес потребує інтеграції передових обчислювальних методологій, використання сучасної обчислювальної майстерності для навігації в складнощах, притаманних ідентифікації різноманітних і складних систем [61]. Ключова роль, яку відіграє ідентифікація в царині автоматизації, підкреслює її незамінність, слугуючи фундаментом для побудови адаптивних, стійких і цілеспрямованих систем керування серед складного гобелена сучасних промислових ландшафтів.

Ідентифікація – це процес побудови математичної моделі об'єкта, дуже подібний об'єкту з точністю до заданого критерію.

Проблема ідентифікації складається з багатьох аспектів і для її реалізації необхідно вирішити широкий спектр теоретичних та практичних завдань.

В основі перетворення звичайних складів на розумні складські приміщення лежить Cyber-Physical System (CPS) [62].

Cyber Physical system – покоління цифрових систем [63], які представляють собою обчислювальні та фізичні можливості взаємодії, керування та контролю за допомогою комп'ютерних алгоритмів [64].

На логістичних підприємствах досить часто використовують RFID системи. RFID (Radio Frequency IDentification) [7] – автоматична ідентифікація об'єктів, за допомогою радіосигналів зчитуються або записуються дані, що зберігаються в RFID-мітках. Будь-яка RFID-система складається із пристрою зчитування і транспортера (RFID-мітка, RFID-тег), а також системи обліку.

Система обліку – це програми, які зберігають та аналізують інформацію, пов'язуючи елементи в єдину систему.

За дальністю зчитування RFID-системи поділяються на:

- системи ближньої ідентифікації (зчитування на відстані до 20 см);
- системи ідентифікації середньої дальності (від 20 см до 5 м);
- дальньої ідентифікації (від 5 м до 300 м).

RFID-мітка – це запам'ятовуючий пристрій, який складається з двох частин, інтегральної схеми для зберігання й обробки інформації та антени для прийому та передачі сигналу [7]. Мітка зберігає унікальні інформаційні дані та номер. На рис. 1.5 зображено приклад RFID-мітки.

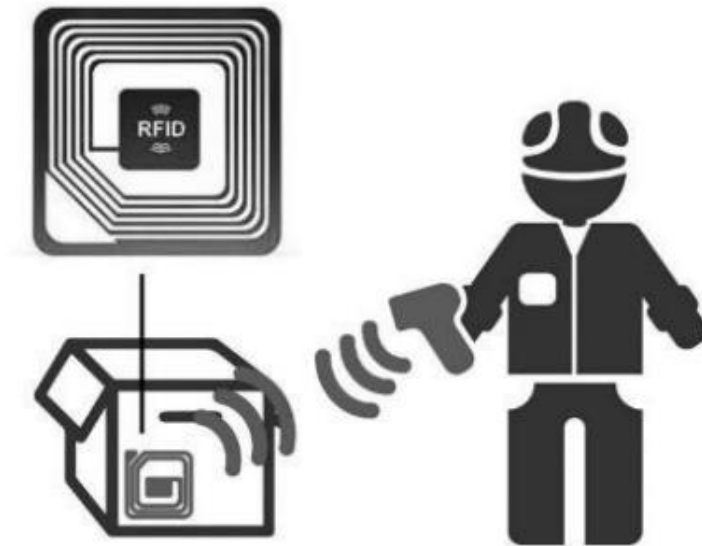


Рисунок 1.5 – Приклад RFID-мітки

Системи RFID (радіочастотної ідентифікації) служать багатогранним цілям у кількох областях, зокрема у виробництві, логістиці, зберіганні, бібліотеках тощо. У виробничих умовах ці системи є інструментальними [19] для реєстрації та керування сировиною, готовою продукцією, запасами та різними компонентами. Вони оптимізують логістику та складські операції, забезпечуючи точне відстеження руху товарів, тим самим прискорюючи процеси сортування.

У бібліотеках мітки RFID революціонізують управління книгами, сприяючи ефективній організації книжкових полиць і системам видачі та пошуку книг. Численні переваги використання систем RFID охоплюють кілька ключових особливостей. До них належать здатність перезаписувати дані на етикетці, швидке отримання інформації з етикеток, збільшені відстані зчитування, розміщення значного обсягу даних у кодї (до 10 000 байт), стійкість до зовнішніх факторів разом із високою довговічністю.

Мітки RFID бувають різних форм, визначених контекстами їх передбачуваного застосування [20]. Сюди входять самоклеючі паперові або майларові етикетки, стандартні пластикові картки, етикетки для дисків і формати брелків. Зокрема, RFID-карти, компактні безконтактні пластикові картки з прихованими бирками, що зберігають важливу інформацію, знаходять повсюдне застосування. Вони доступні в різних налаштуваннях, від карток оплати проїзду в громадському транспорті до ідентифікаційних карток компанії та дисконтних карток роздрібних магазинів.

Інтеграція та розгортання систем RFID значно оптимізує роботу в різних

На рис. 1.6 зображено приклад штрих-коду та пристрій для його зчитування.



Рисунок 1.6 – Приклад штрих-коду та пристрій для його зчитування

Навпаки, штрих-коди [21] складаються з послідовних горизонтальних чорних ліній і невеликих білих прогалів, що кодує інформацію у форматі, зручному для читання пристроєм. Їхня функція полягає в пропонуванні ефективних засобів для пошуку даних, що обслуговують різні сектори за допомогою стандартизованих методів кодування для інвентаризації, роздрібною торгівлі та логістики. Системи RFID, завдяки своїй універсальній функціональності та стійкості, представляють собою зміну парадигми в управлінні даними та контролі запасів, революціонізуючи робочі процеси в

галузях промисловості та підвищуючи ефективність пошуку та управління інформацією.

Серед двовимірних штрих-кодів зразками є коди QR (Quick Response). На відміну від традиційних лінійних штрих-кодів, QR-коди мають здатність кодувати значно більші обсяги даних. Ці коди функціонують у двох вимірах, полегшуючи кодування інформації як по горизонтальній, так і по вертикальній осях. На відміну від лінійних штрих-кодів, які вимагають стандартних зчитувачів, для декодування двовимірних кодів потрібні спеціальні сканери або спеціальне програмне забезпечення, спеціально розроблене для їх інтерпретації.

QR-код (quick response – швидке реагування) – як відомий варіант двовимірних штрих-кодів, являють собою динамічне сховище, що містить віртуальну базу даних, яка може вмістити приблизно 4200 символів. Ці коди, доступні через програми на смартфонах або сумісних пристроях, надають миттєвий доступ до інформації, що зберігається в їхніх унікальних наборах символів.

Їх широке застосування поширюється, зокрема, на промисловій сфері, служачи інструментальними інструментами для розпізнавання та ідентифікації об'єктів у промислових екосистемах. Кожна товарна одиниця у виробництві або кінцевий продукт прикріплюється QR-кодом [2], що містить спектр відповідної інформації. Ці дані можуть включати такі важливі деталі, як номери моделі чи серійні номери, дати виробництва, терміни придатності (що стосується харчової промисловості), цільове призначення, адреси призначення (у контексті логістики) тощо. Універсальність і адаптивність QR-коду роблять його основою в сучасній промисловій практиці, сприяючи оптимізованому управлінню запасами та сприяючи ефективному відстежуванню у сферах виробництва, логістики та споживачів. Інтеграція QR-кодів у промисловість [21] не тільки прискорює пошук інформації та ідентифікацію об'єктів, але й служить каналом для безперервного обміну даними, пропонуючи підвищену прозорість і точність у логістиці ланцюга поставок. Їхнє застосування є прикладом зміни парадигми в асиміляції цифрових технологій у промислових рамках, відстоюючи

ефективність, точність і доступність відповідних даних у реальному часі протягом життєвого циклу виробництва.

На рис. 1.7 представлено приклад QR-коду та його зчитування.



Рисунок 1.7 – Приклад QR-коду та його зчитування

Оптимізація матеріально-технічного забезпечення зберігання різноманітних запасів вимагає управління та зберігання великих обсягів даних. Впровадження QR-коду являє собою автоматизоване рішення для ідентифікації об'єктів, що обіцяє суттєве зниження помилок обліку запасів. Використання технології QR-кодів стає надійною методологією, що забезпечує точність у постійному управлінні та нагляді за складськими запасами. Його інтеграція обіцяє підвищену точність і ефективність у рамках контролю та управління складом, проголошуючи надійний шлях для безперервних операційних процесів із мінімізованою кількістю помилок.



## Висновки до розділу

Загальний висновок щодо автоматизації процесів конвеєрного сортування полягає в тому, що впровадження сучасних технологій автоматизації має великий потенціал для оптимізації та підвищення ефективності виробничих процесів у різних галузях промисловості. Високий рівень точності, швидкості та надійності автоматизованого сортування робить його надзвичайно привабливим для компаній, які прагнуть оптимізувати свої робочі процеси та скоротити витрати.

Автоматизоване сортування на конвеєрі значно зменшує вплив людини на процес, що призводить до зменшення кількості помилок і відходів, а також до підвищення загальної продуктивності. Використання передових систем машинного навчання і штучного інтелекту дозволяє автоматично адаптувати процес сортування до мінливих умов і підвищує його гнучкість, забезпечуючи оптимальну реакцію на зміни у виробничому середовищі.

Однак автоматизація конвеєрного сортування вимагає значних витрат на впровадження та обслуговування спеціалізованого обладнання та програмного забезпечення. Більшість компаній повинні ретельно оцінити переваги та витрати, пов'язані з впровадженням автоматизованих систем, перш ніж прийняти рішення про їх використання.

Успішне впровадження автоматизованих процесів конвеєрного сортування значною мірою залежить від правильної стратегії впровадження, навчання персоналу та підтримки системи.

## 2 РОЗРОБКА СТРУКТУРИ СИСТЕМИ

### 2.1 Апаратне забезпечення

В контексті розробки системи сортування об'єктів за QR-кодом, важливим аспектом є вибір оптимального апаратного забезпечення для забезпечення ефективності та точності процесу. Розробка такої системи передбачає обґрунтований вибір апаратних компонентів, які забезпечать необхідну швидкодію та стабільність у роботі.

Конвеєр, призначений для сортування об'єктів за допомогою QR-сканера, рис. 2.1, складається з декількох ключових елементів, починаючи з точки старту, де об'єкти починають свій шлях по конвеєру. QR-сканер використовується для зчитування кодів на об'єктах, що дозволяє ідентифікувати їх та визначати напрямок подальшого руху. Коли об'єкт проходить через QR-сканер, інформація зчитується і передається до контролера, який відповідає за подальші дії конвеєра.

Зштовхувачі розташовані на лінії конвеєра та функціонують як механізми, що перенаправляють об'єкти на різні лінії в залежності від інформації, отриманої від QR-сканера. Вони активуються контролером та забезпечують рух об'єктів відповідно до програмованих параметрів. Ці зштовхувачі мають зону помилки, де об'єкти можуть бути направлені у випадку невдачі зчитування чи неправильної ідентифікації QR-кодів.

Контролер конвеєра відповідає за координацію руху об'єктів на основі інформації від QR-сканера. Він регулює роботу зштовхувачів, вирішує, куди направити об'єкт після сканування, та керує процесом сортування. Контролер також має можливість перенаправити об'єкт у зону помилки для повторного сканування або виправлення помилок ідентифікації.

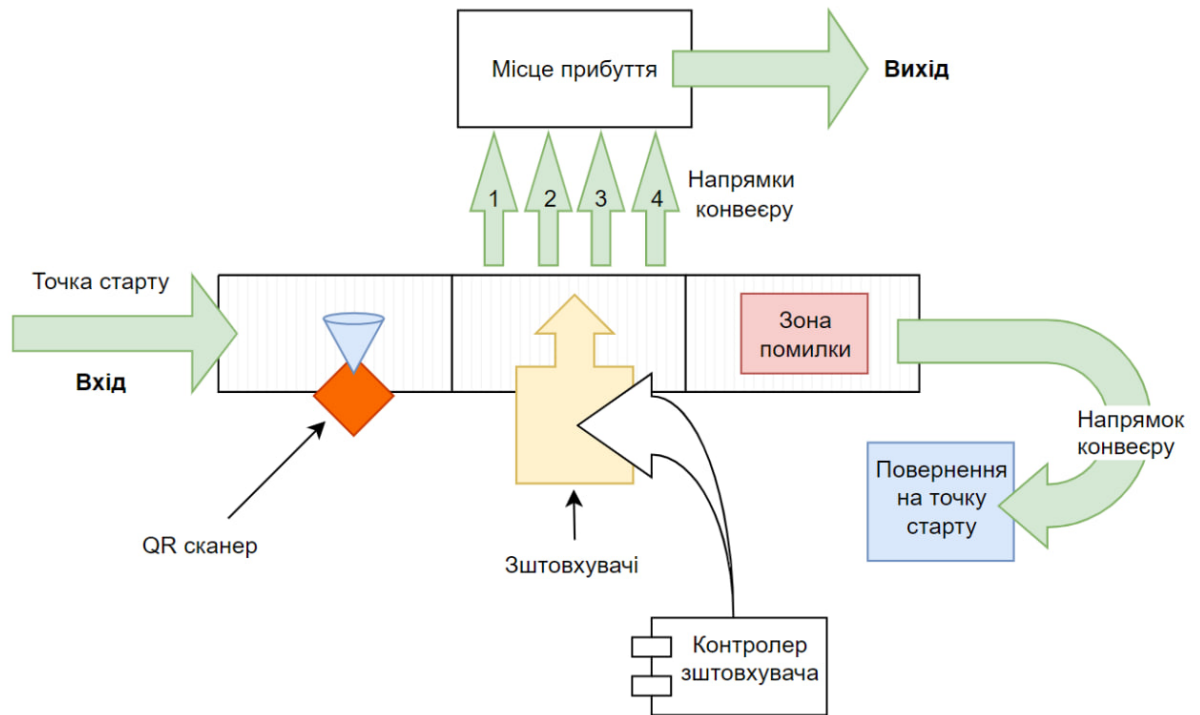


Рисунок 2.1–Схема роботи системи

Місце прибуття на конвеєрі призначене для об'єктів, які успішно пройшли сканування та сортування. Це кінцева точка для правильно ідентифікованих об'єктів, які готові до подальшого використання або відправлення у відповідне місце призначення.

Точка закінчення конвеєра представляє собою остаточну зупинку для оброблених об'єктів. Після успішного сортування та проходження через місце прибуття об'єкти направляються до точки закінчення, де вони можуть бути взяті для подальшої обробки, використання або відправлення.

У цілому, конвеєр із QR-сканером є складною системою, рис. 2.2, яка поєднує в собі технологію сканування з механізмами сортування та контролю руху об'єктів з метою ефективного та точного розподілу матеріалів чи товарів.

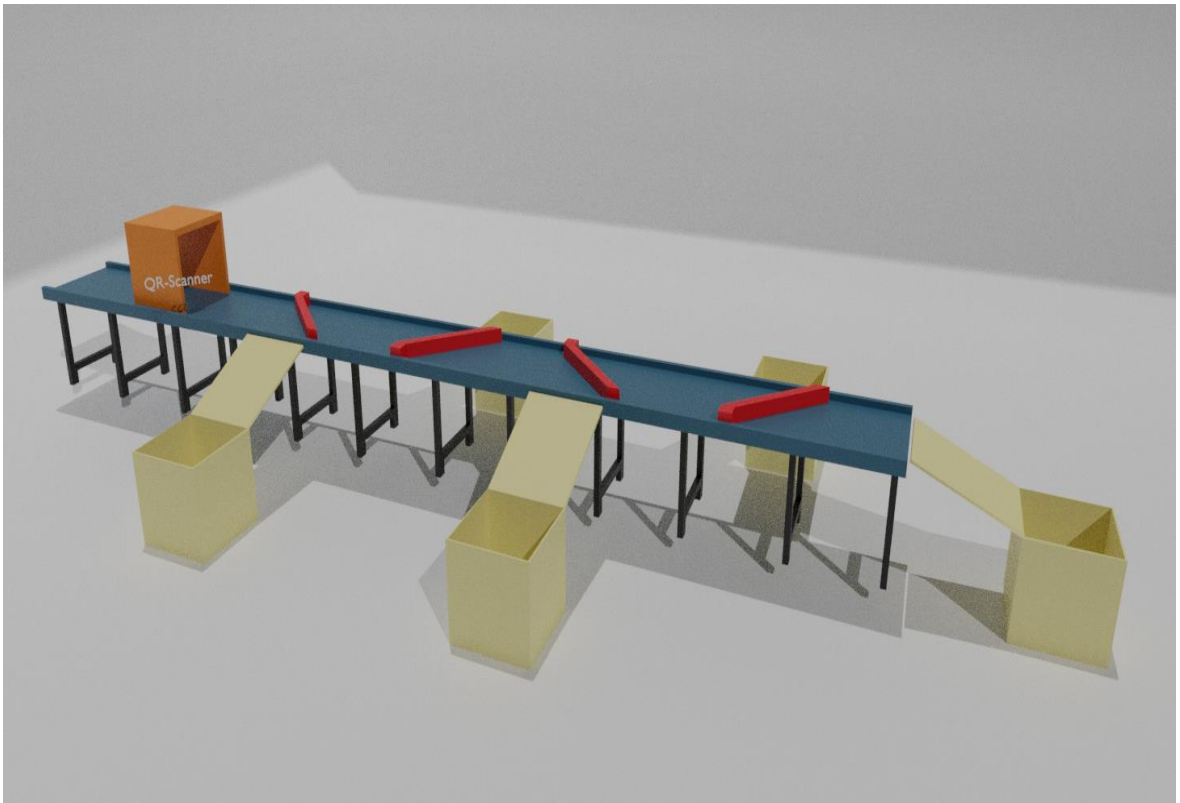


Рисунок 2.2– Загальний вигляд роботи системи

Система складається з контролера, який керує системою, камер з датчиками, конвеєра та зіштовхувача типу «Push Tray Sorter».

ПК або вбудовані системи (embedded systems): можуть використовуватися для більш складних завдань управління, обробки даних та комунікації з іншими системами в реальному часі.

Комп'ютер (ПК), використовуваний у системі сортування за QR-кодом на конвеєрі, може бути центральним управляючим пристроєм, що координує всі компоненти системи. Ось деталізовані аспекти його функцій та характеристик:

- Управління програмним забезпеченням: ПК використовується для запуску спеціалізованого програмного забезпечення, яке обробляє інформацію зчитаних QR-кодів, приймає рішення щодо маршрутизації продуктів на конвеєрі та керує роботою різних частин системи.
- Обробка даних у реальному часі: ПК забезпечує обробку інформації з датчиків та камер у реальному часі. Він отримує дані від

них, а потім аналізує, інтерпретує та передає команди до компонентів системи.

- Комунікація з іншими системами: ПК може бути інтегрованим з іншими системами автоматизації або управління складом. Це дозволяє обмінюватись даними, такими як інформація про інвентар, статистика процесів тощо.
- Система управління базами даних: ПК може використовуватися для зберігання та управління базами даних з інформацією про товари, шляхи сортування, статистика роботи системи тощо.
- Графічний інтерфейс для оператора: ПК може мати графічний інтерфейс, який дозволяє операторам відстежувати роботу системи, перевіряти процес сортування, відлагоджувати проблеми та вносити зміни в параметри роботи системи.

Комп'ютер в цій системі відіграє роль центрального мозку, який координує всі аспекти сортування на конвеєрі, забезпечуючи ефективну роботу та обробку даних.



Рисунок 2.3– Модуль USB-камери

Модуль USB-камери ELP 8MP рис. 2.3, з автофокусом і датчиком CMOS IMX179 - це високоякісна USB-камера, призначена для захоплення високороздільних зображень. Особливості цього модуля:

- Роздільна здатність 8MP: Ця камера здатна захоплювати зображення в роздільній здатності 8 мегапікселів, що дозволяє отримувати деталізовані знімки об'єктів.
- Датчик CMOS IMX179: Цей датчик забезпечує високу якість зображення та дозволяє камері працювати у різних умовах освітлення.
- Автофокус: Функція автофокусу дозволяє камері автоматично налаштовувати фокус для отримання чіткого зображення об'єкта навіть при зміні відстані до нього.
- Міні-лінза з кутом огляду 76 градусів: Ця лінза забезпечує широкий кут огляду, що дозволяє камері охоплювати більше області для зчитування QR-кодів або для відстеження предметів

Використання модуля USB-камери в системі сортування за QR-кодом на конвеєрі дозволяє точно зафіксувати QR-коди на продуктах або предметах, що проходять через систему. Висока роздільна здатність, якість датчика та можливість автоматичного фокусування забезпечують точне і надійне зчитування QR-кодів навіть на великій швидкості конвеєра. Широкий кут огляду дозволяє камері охоплювати більше області, забезпечуючи ефективне сканування кодів навіть при рухомих об'єктах.

Зіштовхувач «Push Tray Sorter» - це система сортування, рис. 2.4, яка використовується для розділення продуктів на конвеєрі за їхнім призначенням або іншими критеріями. Ця система має спеціальні лотки (треї), які рухаються вздовж конвеєра і використовуються для відсортування товарів.

Основна ідея полягає в тому, що коли товар досягає точки сортування, зіштовхувач активується та зрушує лоток, відправляючи товар у відповідний напрямок. Цей механізм забезпечує точне розділення товарів за певними критеріями, наприклад, за QR-кодом, розміром, категорією або будь-якою іншою характеристикою, що необхідна для сортування.



Рисунок 2.4– Зіштовхувач «Push Tray Sorter»

Використання зіштовхувача «Push Tray Sorter» у системі сортування дозволяє ефективно та швидко розділяти товари на конвеєрі, дотримуючись певних критеріїв. Це може значно полегшити процес логістики та сортування товарів у складі або на виробництві, підвищуючи продуктивність та точність сортування.

Вибір апаратного забезпечення для цієї системи може включати в себе визначення оптимальних параметрів процесора, обсягу оперативної пам'яті, типу та роздільної здатності камери, а також необхідності наявності додаткових сенсорів чи модулів зв'язку.

Одним із ключових критеріїв при виборі апаратних компонентів є їхні технічні можливості, зокрема, здатність працювати з великим обсягом даних, швидкість обробки та точність зчитування QR-кодів.

Для оптимізації роботи системи важливо підібрати апаратне забезпечення, яке забезпечить оптимальне співвідношення між продуктивністю та витратами енергії.

У процесі вибору апаратних рішень слід ураховувати можливість майбутнього розширення функціональності системи та сумісність з потенційними оновленнями програмного забезпечення.

Додатково, апаратне забезпечення повинне бути здатним забезпечувати високу стабільність роботи системи при різних умовах експлуатації, таких як різні освітлення, розміри та форми об'єктів для сортування.

Під час вибору апаратних компонентів слід також враховувати можливості їхньої інтеграції з існуючим програмним забезпеченням та здатність до взаємодії з іншими пристроями чи системами.

Ефективність роботи системи сортування QR-кодів значно залежить від правильного вибору апаратних компонентів, що вимагає комплексного підходу та врахування широкого спектру технічних характеристик та вимог до функціональності системи [47, 48].

У цілому, вибір апаратного забезпечення для системи сортування за QR-кодом - це складний і багатогранний процес, який потребує уважного аналізу та збалансованого підходу для досягнення оптимальної продуктивності та функціональності системи.

## 2.2 Загальна структура системи

Етап розробки системи концентрується на аналізі та проектуванні певного набору впорядкованих видів діяльності, що здійснюється в рамках кожного проекту з розробки програмного забезпечення та називається – життєвим циклом.

Етап проектування поділяється на два основні підпункти: архітектурне та деталізоване проектування. Життєвий цикл зображено на рис. 2.5.

Архітектурне проектування потрібне для оцінки та аналізу ризиків, пов'язаних з обраною архітектурою. Також визначаються принципи масштабування системи, забезпечуючи її здатність росту та адаптації до змін у



майбутньому. Крім того, на цьому етапі вирішують питання щодо безпеки системи, обираються методи захисту від потенційних загроз.



Рисунок 2.5– Життєвий цикл розробки програмного забезпечення

У процесі деталізованого проектування звертають увагу на оптимізацію роботи системи, враховуючи вимоги до швидкодії та продуктивності. Ведеться розробка документації, яка необхідна для подальшого програмування, тестування та підтримки системи. Крім того, на цьому етапі проводяться детальні внутрішні та зовнішні рецензії проекту для виявлення можливих недоліків та покращень.

Під час деталізованого проектування також вирішують питання інтеграції з іншими системами, визначаються стандарти та протоколи комунікації між компонентами програмного забезпечення. Встановлюються методи тестування для перевірки функціональності та стабільності системи перед його впровадженням.

На обох етапах проектування важливу роль відіграє комунікація між учасниками проекту: програмістами, архітекторами, тестувальниками та

замовниками. Чітке розуміння потреб підприємства дозволяє краще адаптувати архітектуру та деталі реалізації до конкретних вимог та очікувань замовника.

Також на цих етапах важливо враховувати масштаби проєкту та обсяги робіт, щоб забезпечити реалістичні терміни виконання та ресурси, необхідні для успішної реалізації програмного продукту. Крім того, розробники повинні враховувати можливість майбутнього розширення та модернізації системи для відповіді на змінювані потреби користувачів.

Проєктування архітектури системи сортування об'єктів за QR-кодом на базі рушія Unity вимагає детального планування та врахування різних компонентів системи [17].

Важливим аспектом є розробка алгоритмів сортування об'єктів за QR-кодом, що може включати в себе розпізнавання та інтерпретацію інформації з кодів, їх класифікацію за певними критеріями та відповідну обробку.

Під час проєктування вирішуються питання безпеки системи, включаючи захист від можливих атак або зловживань з боку користувачів. Також враховується масштабованість системи, зокрема, її здатність працювати ефективно при збільшенні обсягів даних чи кількості оброблюваних QR-кодів.

Розробка інтерфейсу користувача також є важливим аспектом архітектури, вона включає в себе створення інтуїтивного та зручного інтерфейсу для взаємодії з системою сортування об'єктів за допомогою QR-кодів.

Детальне тестування системи на кожному етапі розробки є необхідною складовою процесу, щоб виявити та усунути можливі помилки або недоліки, що можуть виникнути під час роботи системи.

Завершальним етапом є документування процесу проєктування та розробки, що включає в себе створення технічної документації з описом архітектури, алгоритмів та методів, використаних у системі сортування за QR-кодом.

Такий процес проєктування дозволяє створити ефективну, масштабовану та безпечну систему сортування об'єктів за QR-кодом на базі рушія Unity, відповідну вимогам та потребам користувачів.

Unity - це потужний інструмент для розробки ігор та додатків у віртуальному середовищі. Потрібно вибрати додаткові бібліотеки та плагіни, які дозволять опрацьовувати QR-коди. Для цього можна використовувати бібліотеки, такі як ZXing[24], UnityQRCode[25] або спеціалізовані плагіни для роботи з камерою.

Першим кроком є детальний аналіз вимог до системи. Потрібно визначити, які види QR-кодів будуть використовуватися, яка інформація повинна зберігатися в QR-кодах, які завдання системи сортування повинна вирішувати (наприклад, ідентифікація об'єктів, визначення їхнього місця розташування тощо) [22].

Початковим етапом є аналіз вимог до системи, де враховуються потреби користувачів та функціональні особливості, які повинна виконувати система [24]. Основні кроки включають в себе визначення функціональних та нефункціональних вимог, встановлення ключових властивостей QR-кодів, визначення можливих форматів зберігання та обробки даних. Крім того, важливим етапом є вибір технологій та інструментів, що найбільш відповідають поставленим завданням, таких як бібліотеки для роботи з QR-кодами, фреймворки для роботи з рушієм Unity, тощо.

Далі, у процесі проектування архітектури вирішуються питання, пов'язані зі структурою системи, розподілом функціональності між компонентами, визначенням модулів для обробки QR-кодів, взаємодії з базою даних або зовнішніми сервісами для зберігання та отримання інформації з кодів.

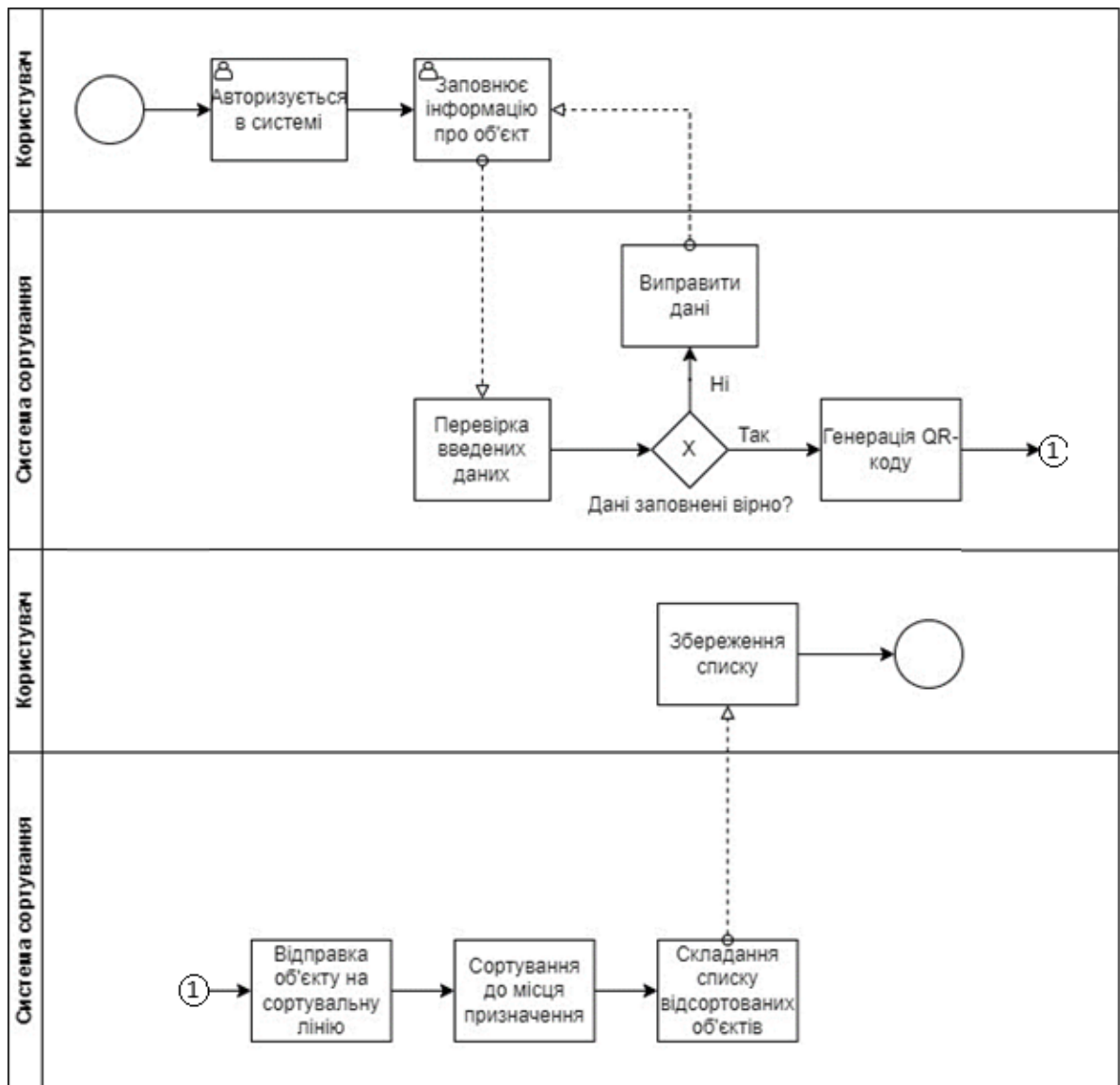


Рисунок 2.6– Діаграма діяльності програмного забезпечення

На рис. 2.6 зображено діаграму діяльності програмного забезпечення, для чого потрібно виконати наступні кроки:

Розробка інтерфейсу для взаємодії з системою сортування. Розробка UI-елементів для відображення інформації з QR-кодів, кнопки для виконання дій, такі як сортування або пошук, та інші компоненти, необхідні для зручного управління системою.

Розробка модуля для зчитування та обробки QR-кодів. Можливо використовувати камеру мобільного пристрою або веб-камеру комп'ютера для зчитування QR-кодів. Після зчитування QR-коду система повинна отримати з нього інформацію та виконати відповідні дії.

Реалізувати логіку сортування на основі інформації, отриманої з QR-кодів. Визначити, як об'єкти повинні бути сортовані та розташовані на основі цієї інформації. Розробити алгоритми для оптимального розміщення об'єктів[22].

Для зберігання даних про об'єкти та їхнє розташування можна використовувати базу даних або локальне сховище даних.

Перед впровадженням системи провести тестування для перевірки правильності зчитування QR-кодів, логіки сортування та коректності роботи інтерфейсу. Відлагодити систему, якщо необхідно.

Після успішного тестування впровадити систему в реальне виробниче або логістичне середовище.

Завершити впровадження системи потрібно забезпечити її підтримку та розглянути можливості подальшого розвитку та оптимізації. Реагувати на потреби та вимоги користувачів для покращення функцій [39].

### 2.3 Алгоритмічне забезпечення

Алгоритмічне забезпечення системи сортування об'єктів за QR-кодом є ключовим елементом у розробці систем, спрямованих на ідентифікацію та класифікацію об'єктів на основі цифрової інформації, закодованої у вигляді QR-кодів. Вивчення та застосування алгоритмів у цьому контексті вимагає глибокого розуміння функціональності QR-кодів, їх структури, а також особливостей об'єктів, які підлягають сортуванню.

Провідними алгоритмами в цій області є алгоритми розпізнавання QR-кодів, які включають в себе стадії зчитування та інтерпретації інформації з кодів. Ці алгоритми базуються на аналізі структури QR-коду, виявленні та розшифруванні інформації, закодованої у вигляді шаблонів і модулів, присутніх у самому коді.

Для ефективності роботи системи сортування за QR-кодом використовують різноманітні алгоритми класифікації об'єктів на основі даних,

отриманих із QR-кодів. Ці алгоритми можуть базуватися на розпізнаванні певних параметрів чи шаблонів у зчитаній інформації та її подальшій інтерпретації для визначення категорій об'єктів.

Додатково, алгоритми сортування за QR-кодом можуть включати в себе методи ранжування об'єктів відповідно до певних критеріїв, які визначаються на основі інформації, представленої у QR-кодах. Це може бути важливим у контексті систем, де необхідно швидко та ефективно впоратися з великим обсягом об'єктів та класифікувати їх за певними параметрами.

У системах сортування за QR-кодом можуть використовуватися алгоритми оптимізації для забезпечення швидкої та ефективної обробки великого обсягу даних, які містяться у QR-кодах. Ці алгоритми спрямовані на підвищення продуктивності системи та оптимізацію ресурсів, необхідних для обробки та аналізу інформації.

Алгоритмічне забезпечення системи сортування об'єктів за QR-кодом передбачає розробку та використання алгоритмів, які здатні ефективно та точно сортувати значну кількість об'єктів за QR-кодами. Для цього використовуються різні алгоритмічні підходи, які враховують такі фактори, як розмір коду, тип даних, швидкодія та ефективність сортування.

Один з ключових алгоритмів, які застосовуються для сортування об'єктів за QR-кодом, - алгоритм "QuickSort" [58]. Його робота зображена на рис. 2.7. Цей алгоритм базується на стратегії розділення масиву на менші підмасиви та порівнянні елементів. Він є швидким та ефективним для сортування великого обсягу даних.

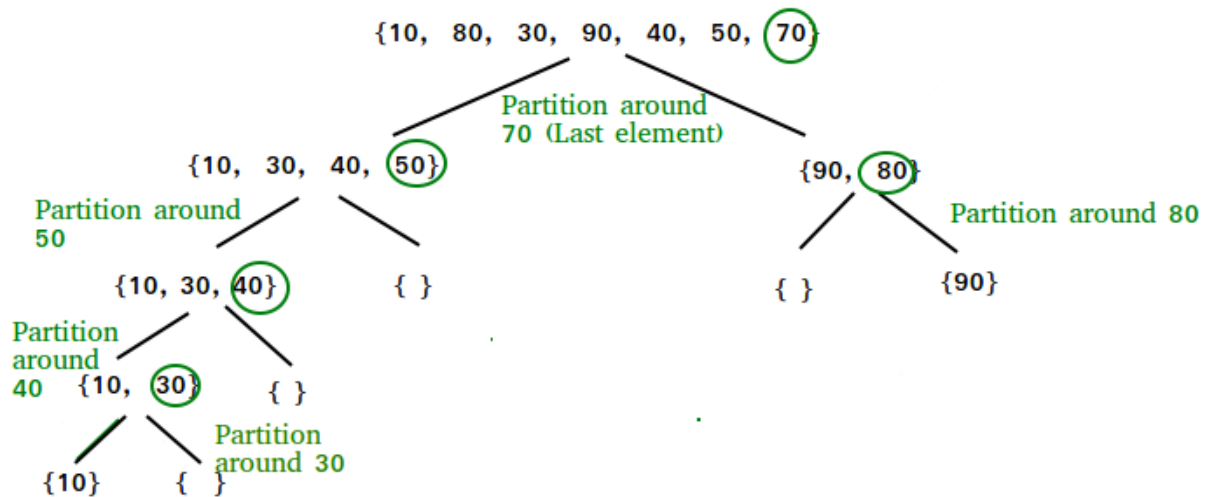


Рисунок 2.7– Як працює алгоритм "QuickSort"

Інший важливий алгоритм - "MergeSort". Він базується на розбитті масиву на менші підмасиви, сортуванні їх окремо і потім злитті відсортованих підмасивів. Цей алгоритм гарантує стабільність сортування і має складність  $O(n \log n)$ . Рисунок 2.8 наводить етапи роботи алгоритму "MergeSort".

Крім "QuickSort" та "MergeSort", існує ряд інших алгоритмів сортування, які можуть бути використані для сортування об'єктів за QR-кодом. Деякі з них, такі як "HeapSort" та "BubbleSort", використовуються для швидкого сортування, тоді як інші, наприклад "InsertionSort" та "SelectionSort", підходять для менших обсягів даних або в ситуаціях, коли час виконання не є критичним.

Практичне застосування алгоритмічного забезпечення системи сортування об'єктів за QR-кодом включає організацію швидкого пошуку та ідентифікації об'єктів у великих масивах даних. В логістичних системах, наприклад, QR-коди використовуються для ідентифікації товарів, а алгоритми сортування допомагають швидко знаходити необхідні товари на складі.

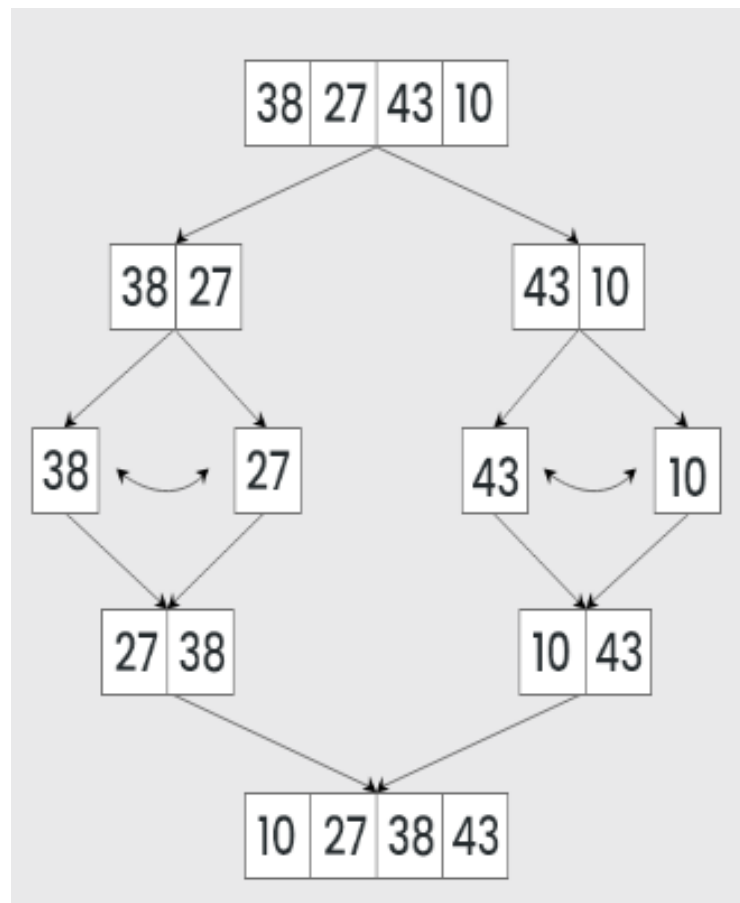


Рисунок 2.8– Як працює алгоритм "MergeSort"

Наукові дослідження в галузі алгоритмічного забезпечення системи сортування об'єктів за QR-кодом спрямовані на вдосконалення існуючих алгоритмів та розробку нових підходів для оптимізації процесу сортування. Вчені працюють над розробкою алгоритмів, які забезпечують високу швидкодію, надійність та точність сортування великих обсягів даних за QR-кодами.

Алгоритмічне забезпечення системи сортування об'єктів за QR-кодом становить суттєву складову цілісного функціонування та ефективної роботи системи, що спрямована на класифікацію та ідентифікацію об'єктів на основі цифрової інформації, закодованої у вигляді QR-кодів.

Час виконання кожного оператора залежить від середовища виконання, операційної системи та інших системних характеристик.

Залежно від ефективності існує багато типів алгоритмів, серед яких можна виділити такі типи алгоритмів зображені на рис. 2.9.



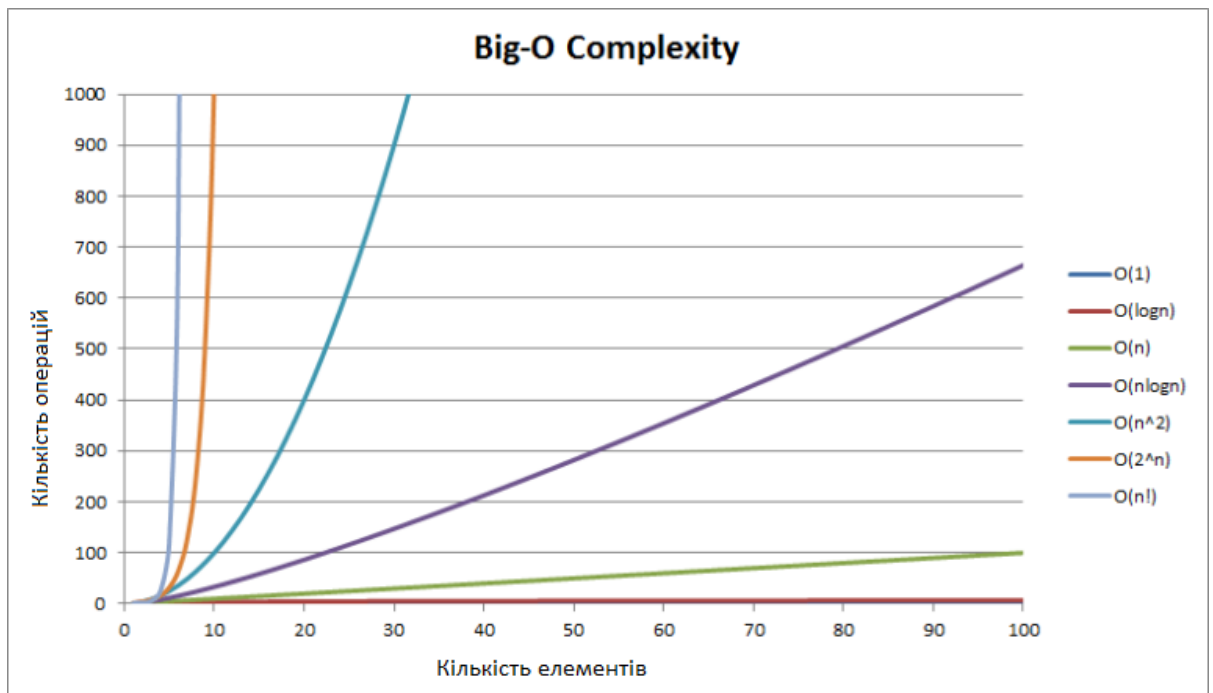


Рисунок 2.9– Модель відношення складності алгоритму

Особливу увагу в алгоритмічному забезпеченні системи сортування за QR-кодом приділяють виявленню та усуненню можливих помилок у розпізнаванні чи інтерпретації інформації з кодів. Це включає в себе розробку алгоритмів корекції помилок та адаптації до різних умов зчитування QR-кодів.

Використання алгоритмів у системах сортування за QR-кодом також вимагає врахування аспектів безпеки, зокрема, захисту від можливих атак, зловживань або фальсифікації інформації, яка міститься у QR-кодах.

У висновку, алгоритмічне забезпечення системи сортування об'єктів за QR-кодом є актуальною та перспективною галуззю досліджень. Шляхом розробки та оптимізації алгоритмів сортування, була проведена робота над вдосконаленням процесу сортування об'єктів за QR-кодом з метою покращення ефективності та точності цього процесу у різних сферах діяльності.

Розробка алгоритмів ідентифікації в системі сортування об'єктів за QR-кодом, рис. 2.10, є ключовим етапом, оскільки вони визначають, як об'єкти будуть ідентифікуватися на основі QR-кодів.

Зчитування QR-коду: Першим кроком є зчитування QR-коду за допомогою камери. Для цього можна використовувати спеціалізовані бібліотеки, такі як

ZXing або UnityQRCode. При зчитуванні QR-коду отримується текстова або бінарна інформація, закодована в QR-коді.

Розкодування інформації: Отриману інформацію з QR-коду потрібно розкодувати, щоб отримати корисні дані. QR-коди можуть містити текст, числа, URL-адреси або іншу інформацію. Алгоритм розкодування повинен враховувати формат і тип даних, закодованих у QR-коді.

Ідентифікація об'єктів: На основі інформації, отриманої з QR-коду, система повинна ідентифікувати відповідний об'єкт або елемент, який відповідає даному QR-коду. Це може бути товар на складі, пакет для доставки або інший об'єкт. Ідентифікація може включати в себе пошук в базі даних або співставлення інформації з QR-коду зі списком об'єктів.

Виконання дій: Після ідентифікації об'єкта система повинна визначити, які дії повинні бути виконані з цим об'єктом. Наприклад, якщо це товар на складі, то може бути потрібно визначити його місце розташування або виконати інші операції, такі як переміщення або підготовка до доставки.

Збереження та оновлення інформації: Після ідентифікації і виконання дій, пов'язаних з об'єктом, система може зберегти або оновити інформацію про стан об'єкта. Ця інформація може використовуватися для подальшого сортування та відстеження.

Обробка помилок і винятків: Розробити механізми для обробки помилок і винятків, які можуть виникнути під час ідентифікації та обробки QR-кодів. Наприклад, якщо QR-код пошкоджений або не може бути розкодований, система повинна вміти виявити це і здійснити відповідну дію.

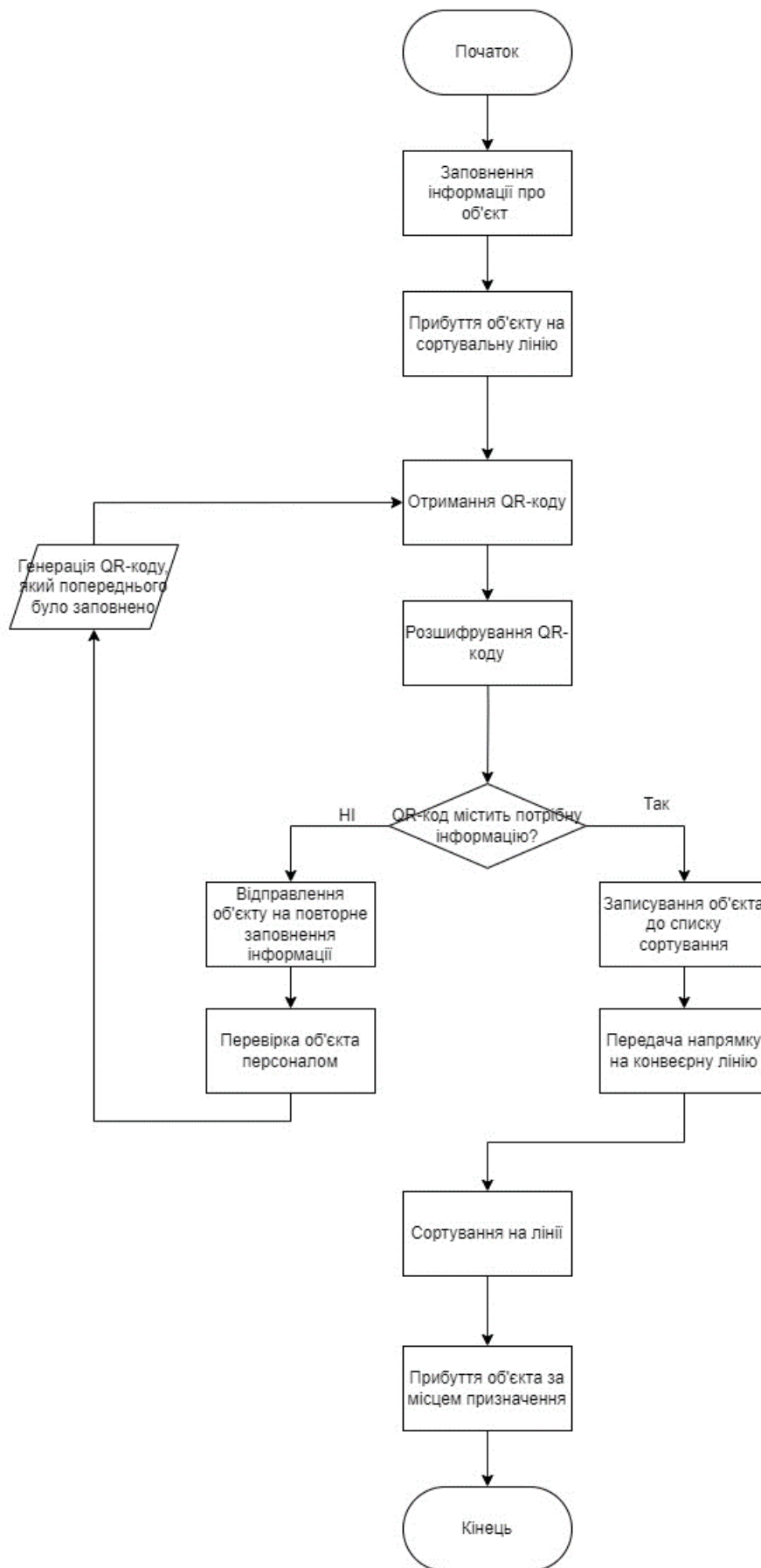


Рисунок 2.10 – Алгоритм ідентифікації

## 2.4 Програмне забезпечення

При розробці системи сортування об'єктів за QR-кодом є багато мов програмування та інструментів, які можуть бути використані, залежно від вимог та уподобань. Ось деякі з них:

**Java:** є популярною мовою програмування для мобільних додатків на платформі Android. Ви можете створювати додатки для зчитування QR-кодів та обробки даних на мові Java [26].

**Python:** це мова програмування з простим синтаксисом, яка підходить для розробки широкого спектру додатків. Ви можете використовувати бібліотеки, такі як `pyzbar` або `OpenCV`, для роботи з QR-кодами [24].

**C#:** це мова програмування, яка часто використовується для розробки додатків для платформи Unity, включаючи AR-додатки для розпізнавання QR-кодів та сортування об'єктів [25].

**JavaScript:** може використовуватися для розробки веб-додатків, які взаємодіють з камерою пристрою для зчитування QR-кодів [26].

**C++:** це мова програмування, яка часто використовується для розробки швидкодіючих додатків та обробки великих обсягів даних. Вона може бути корисною для оптимізації алгоритмів сортування [27].

**Swift:** це мова програмування для розробки додатків для iOS, яка може використовуватися для створення додатків для зчитування QR-кодів [28].

**PHP:** може використовуватися для розробки серверної частини системи, яка обробляє та зберігає інформацію про об'єкти та їхнє розташування [29].

**Node.js:** це середовище виконання JavaScript, яке може використовуватися для створення серверних додатків для обробки запитів, пов'язаних з QR-кодами [30].

**SQL:** Для збереження та управління даними про об'єкти та їхнє розташування ви можете використовувати SQL-системи управління базами даних, такі як MySQL, PostgreSQL або Microsoft SQL Server [31].

**Бібліотеки та фреймворки:** Крім мов програмування, ви можете використовувати спеціалізовані бібліотеки та фреймворки для обробки QR-кодів, такі як ZXing для Java або pyzbar для Python.

Ось порівняння різних мов програмування та інструментів для реалізації системи сортування об'єктів за QR-кодом, включаючи їхні переваги та недоліки, представлено в таб. 2.1.

Таблиця 2.1 – Порівняння сучасних мов програмування

Назва	Переваги	Недоліки
<b>Java</b>	<ul style="list-style-type: none"> <li>• Підтримується на багатьох платформах, включаючи Android.</li> <li>• Має багато бібліотек для роботи з QR-кодами.</li> <li>• Відома своєю надійністю та великою спільнотою розробників.</li> </ul>	<ul style="list-style-type: none"> <li>• Вимагає компіляції, що може затримати розробку.</li> <li>• Може бути менш зручною для швидкого прототипування порівняно з іншими мовами, такими як Python.</li> </ul>
<b>Python</b>	<ul style="list-style-type: none"> <li>• Простий синтаксис і швидке у використанні. Існує багато бібліотек для роботи з QR-кодами, такі як pyzbar і Qrcode.</li> <li>• Зручно використовувати для обробки даних і аналізу.</li> </ul>	<ul style="list-style-type: none"> <li>• Вимагає компіляції, що може затримати розробку.</li> <li>• Може бути менш зручною для швидкої розробки порівняно з іншими мовами, такими як Python.</li> </ul>

Продовження таблиці 2.1 – Порівняння сучасних мов програмування

Назва	Переваги	Недоліки
<b>C# (Unity):</b>	<ul style="list-style-type: none"> <li>• Ідеально підходить для розробки AR-додатків для зчитування QR-кодів.</li> <li>• Має велику спільноту розробників і багато ресурсів для вивчення.</li> <li>• Підтримує розробку на різних платформах.</li> </ul>	<ul style="list-style-type: none"> <li>• Вимагає наявності Unity, що може бути обмеженням в деяких випадках.</li> <li>• Може бути складніше вивчити порівняно з іншими мовами..</li> </ul>
<b>JavaScript</b>	<ul style="list-style-type: none"> <li>• Зручно для розробки веб-додатків з функціональністю QR-кодів.</li> <li>• Підходить для взаємодії з камерою пристроїв через веб-браузери.</li> </ul>	<ul style="list-style-type: none"> <li>• Обмежена функціональність порівняно з мовами, які дозволяють розробляти нативні додатки.</li> <li>• Може вимагати великої кількості JavaScript-коду для реалізації складних функцій.</li> </ul>
<b>C++:</b>	<ul style="list-style-type: none"> <li>• Швидка мова програмування, ідеальна для обробки великих обсягів даних.</li> <li>• Може бути використана для розробки швидкодіючих алгоритмів сортування та обробки QR-кодів.</li> </ul>	<ul style="list-style-type: none"> <li>• Має складний синтаксис і може бути менш доступним для початківців.</li> <li>• Вимагає обережного управління пам'яттю.</li> </ul>

## Висновки до розділу

У цьому розділі роботи було розглянуто різні технології для розробки системи сортування об'єктів за QR-кодами. Було проведено порівняння переваг і недоліків таких технологій, як C# (Unity), Python, JavaScript, C++, Swift та інші. Загалом, вибір технології для розробки системи сортування об'єктів за QR-кодами повинен бути здійснений на основі конкретних потреб та вимог проєкту.

Технологія C# в контексті розробки системи сортування за QR-кодами на платформі Unity виявилася потужним інструментом. Unity надає можливість створювати кросплатформенні додатки, включаючи AR-додатки для розпізнавання QR-кодів, та має велику спільноту розробників. Мова програмування C# є ефективною та надійною, а інтегроване середовище розробки Unity спрощує процес створення інтерактивних додатків [45].

Незважаючи на переваги C# та Unity, вибір технології також залежить від специфіки проєкту і ваших власних знань та вмінь. Важливо ретельно обговорити потреби та вимоги до проєкту з командою розробників та вибрати технологію, яка найкраще відповідає поставленим завданням.

Також було розглянуто життєвий цикл програмного забезпечення та розроблений алгоритм ідентифікації.

## 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Програмне забезпечення “Unity”

Технологія C# в контексті розробки системи сортування об'єктів за QR-кодом з використанням Unity є потужним інструментом для створення інтерактивних додатків, включаючи AR-додатки, які можуть зчитувати QR-коди та взаємодіяти з реальним світом [37].

Unity - це інтегроване середовище розробки (IDE) та фреймворк для створення ігор та візуальних додатків. Він підтримує багато платформ, включаючи Windows, macOS, Android, iOS, Xbox, PlayStation і багато інших. Unity дозволяє створювати кросплатформенні додатки [47], що робить його ідеальним вибором для розробки системи сортування за QR-кодом, яка може працювати на різних пристроях [46].

У цьому контексті Unity виступає як інтегроване середовище розробки (IDE) та фреймворк [54], що надає широкі можливості для створення програмного забезпечення, сумісного з різними платформами.

Інтегроване середовище розробки Unity має широкий спектр інструментів, що спрощують процес створення програм та ігор, від розробки до впровадження. Від інтерфейсу користувача до програмування, цей фреймворк надає різноманітні можливості для ефективної розробки продуктів.

Однією з ключових переваг Unity є його здатність створювати продукти, які можуть працювати на різних пристроях, не обмежуючи їх функціональності або продуктивності[54]. Ця універсальність робить його ідеальним вибором для розробки систем сортування за QR-кодом, які мають працювати на різних пристроях та платформах.

Unity забезпечує високу стабільність та надійність роботи програм, що сприяє впровадженню різноманітних проектів на різних пристроях. Його потужність у поєднанні з можливістю кросплатформенної [56] сумісності робить



його інструментом вибору для розробки та впровадження широкого спектру програм та ігор.

```

1  using System;
2  using System.IO;
3
4  class FileAlphabeticalSort
5  {
6      // Метод для сортування рядків за алфавітом
7      static void SortLinesAlphabetically(string filePath, string newFilePath)
8      {
9          try
10         {
11             // Читання усього вмісту файлу
12             string[] lines = File.ReadAllLines(filePath);
13
14             // Сортування рядків за алфавітом
15             Array.Sort(lines, StringComparer.CurrentCulture);
16
17             // Запис відсортованого вмісту у новий файл
18             File.WriteAllLines(newFilePath, lines);
19
20             Console.WriteLine("Файл був відсортований та перезаписаний з новою назвою.");
21         }
22         catch (Exception e)
23         {
24             Console.WriteLine("Виникла помилка: " + e.Message);
25         }
26     }
27
28     static void Main()
29     {
30         string path = "test.txt";
31         string newPath = "sortedTest.txt";
32
33         // Виклик методу сортування рядків у файлі за алфавітом з перезаписом у новий файл
34         SortLinesAlphabetically(path, newPath);
35     }
36 }
37

```

Рисунок 3.1–Вікно інтегрованого середовища розробки програмного забезпечення «Visual Studio»

Мова програмування C# є офіційною мовою розробки для Unity. Вона відома своєю ефективністю, надійністю та широким спектром бібліотек і інструментів. C# є об'єктно-орієнтованою мовою програмування зі сильною типізацією, що дозволяє розробникам створювати структурований і простий у читанні код [48].

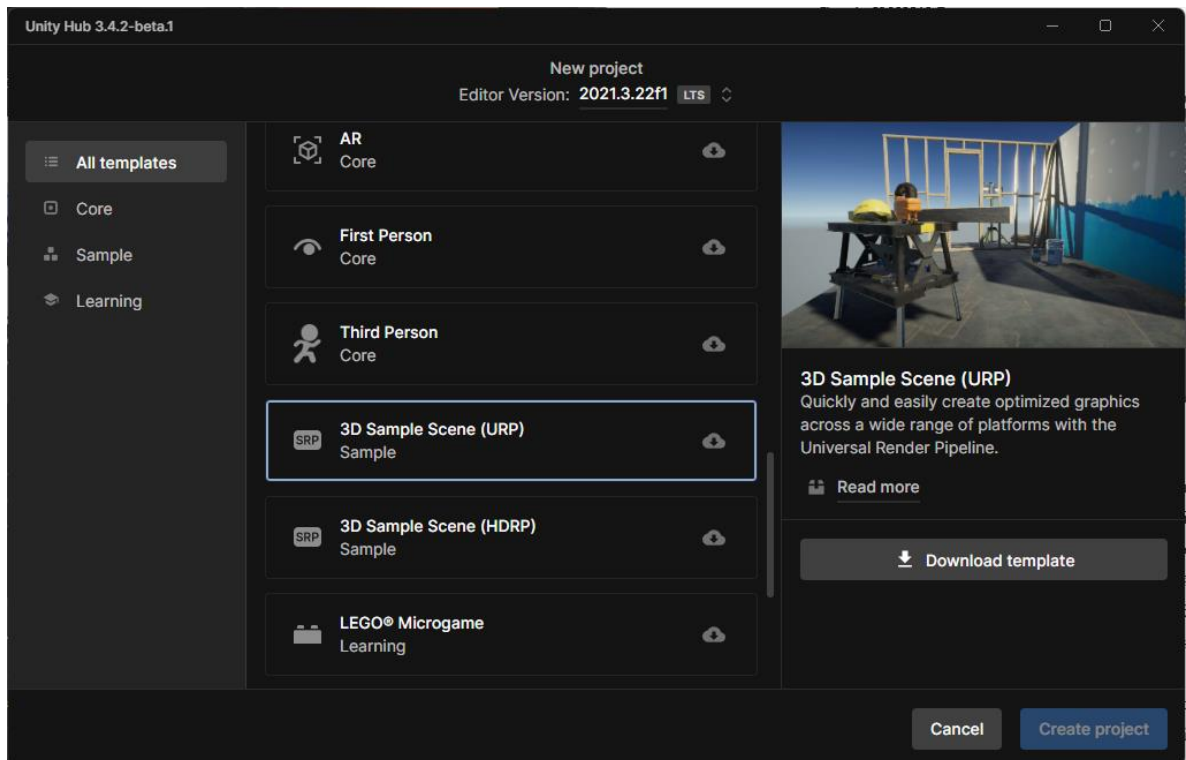


Рисунок 3.2 – Вікно додатку «Unity»

Багатофункціональність: Unity надає розробникам можливість створювати графічно багаті та інтерактивні інтерфейси, додавати анімацію та спеціальні ефекти, а також використовувати фізичний двигун для моделювання поведінки об'єктів. Це дає можливість створити відмінний користувацький досвід [44].

Комплексні проектні засоби: Unity надає інтегровані засоби розробки, такі як візуальний редактор сцен, менеджер ресурсів і система компонентів, що полегшує розробку та управління проектом.

Спільнота та ресурси: Unity має велику активну спільноту розробників, а також багато онлайн-ресурсів, таких як документація, навчальні відео та форуми підтримки.

Недоліки використання Unity та C# для реалізації системи сортування за QR-кодами можуть включати [45]:

- великий розмір додатку, особливо для мобільних пристроїв.
- навчання та знання C# може вимагати часу, якщо ви не знайомі з цією мовою.

- іноді потрібно додаткові ресурси для оптимізації продуктивності на різних платформах.

Загалом, використання Unity та C# може бути дуже ефективним вибором для створення системи сортування об'єктів за QR-кодом/

### 3.2 Розробка генератора QR-коду

Генератор визначає розмір коду в залежності від обсягу інформації, яку необхідно закодувати, а також рівня корекції помилок, що забезпечує стійкість до пошкоджень. QR-код складається з чорних квадратів, що розташовані на білому фоні, і має в собі внутрішню структуру, що містить інформацію у вигляді бінарних даних.

Завдяки своїй специфікації, QR-код може зберігати різноманітні дані, включаючи текст, URL-адреси, контактну інформацію, Wi-Fi налаштування, географічні координати тощо. Генератор використовує внутрішні алгоритми для розташування цих даних у певному порядку, забезпечуючи структурованість та читабельність QR-коду.

Для реалізації генератора QR-коду було обрано ZXing (Zebra Crossing) - це бібліотека для розпізнавання штрих-кодів і QR-кодів. У контексті Unity, існує реалізація ZXing, яка дозволяє використовувати цю бібліотеку для роботи з штрих-кодами та QR-кодами в іграх та додатках, розроблених на цій платформі.

ZXing для Unity пропонує доступ до функцій розпізнавання штрих-кодів і QR-кодів безпосередньо в середовищі розробки Unity. Ця бібліотека дозволяє вбудовувати функціонал розпізнавання кодів безпосередньо в код, що дає нові можливості. Інтеграція бібліотеки ZXing зображена на рис. 3.3.

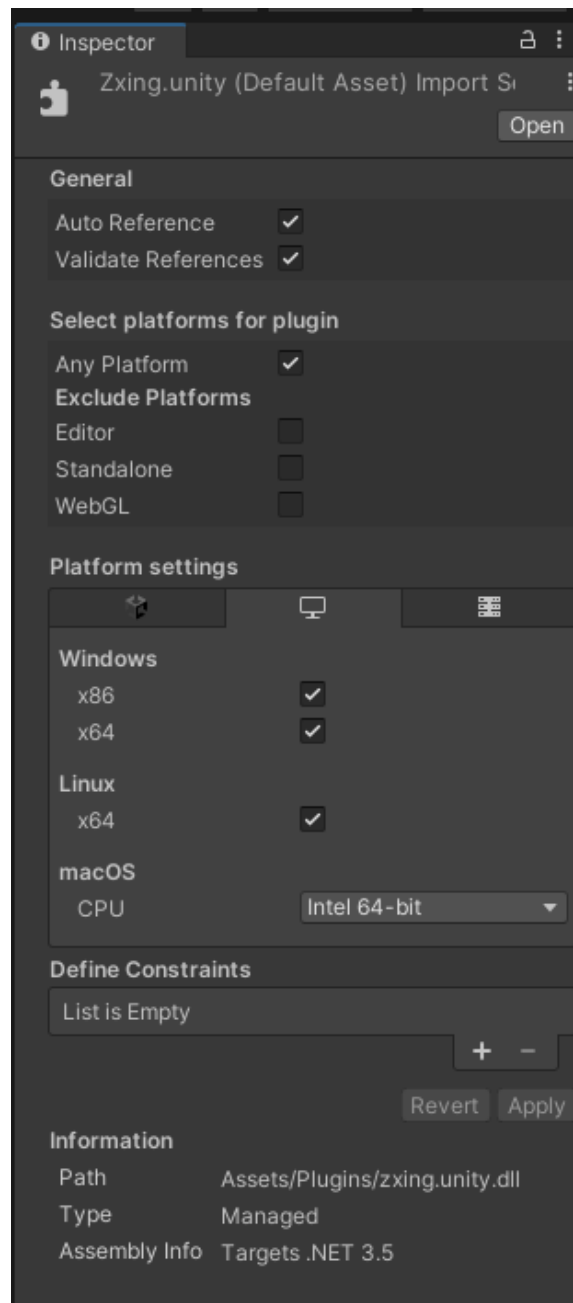


Рисунок 3.3 – Встановлюємо розширення Zxing

Завдяки ZXing для Unity можна легко розпізнавати штрих-коди та QR-коди через вбудовані функції, що дозволяє ігровому двигуну Unity звертатися до камери, знімати зображення, а потім використовувати бібліотеку ZXing для аналізу отриманого зображення та витягнення інформації з коду.

Реалізація ZXing для Unity спрощує процес розпізнавання кодів, дозволяючи розробникам легко використовувати цю функціональність в їхніх проектах без необхідності вдаватися до складних алгоритмів розпізнавання кодів з самого початку.

Такі функції роблять ZXing для Unity потужним інструментом для розробників, які прагнуть інтегрувати можливості розпізнавання штрих-кодів і QR-кодів у свої ігри чи додатки, використовуючи зручний і простий у використанні інтерфейс Unity.

### 3.2.1 Розробка інтерфейсу генератора QR-коду

При розробці інтерфейсу для генерації QR-коду у середовищі Unity, враховувано основні принципи дизайну користувацького інтерфейсу для забезпечення зручності та ефективності користувачів. Цей інтерфейс містить елементи керування, що спрощують процес створення QR-коду, включаючи поля для введення даних, які закодовані, та кнопку для запуску процесу генерації коду.

Інтерфейс інтуїтивно зрозумілий, з чітко розміщеними елементами керування, такими як текстові поля для введення даних, а також кнопки для початку генерації QR-коду.

Головне - це ергономіка і зручність розміщення елементів інтерфейсу на екрані для оптимального використання простору та забезпечення легкості взаємодії з додатком.

Оптимальне розташування елементів інтерфейсу, їх розмір та взаємодія мають забезпечити користувачеві можливість безперешкодно вводити дані, вибирати параметри генерації QR-коду та отримувати результат, що задовольняє його потреби.

Користувацький інтерфейс зображено на рис. 3.4.

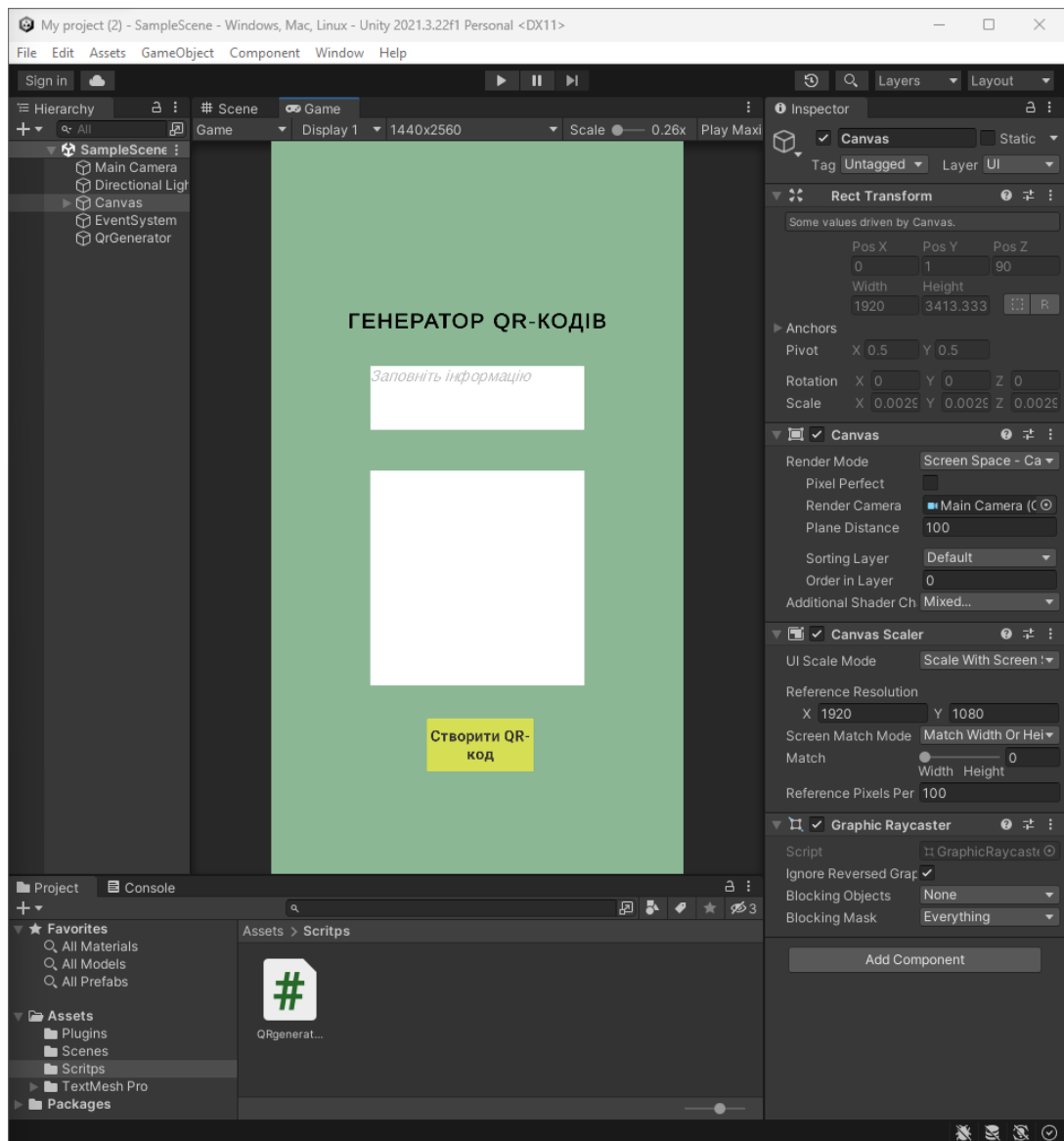


Рисунок 3.4 – Розмітку майбутнього додатку генератора

Важливо пам'ятати про адаптивність інтерфейсу для різних розмірів екранів та пристроїв, щоб забезпечити коректне відображення елементів інтерфейсу на різних платформах та пристроях, що сприяє універсальності додатку.

Важливо враховувати принцип доступності, забезпечуючи можливість користування інтерфейсом для широкого спектру користувачів, включаючи тих, у кого можуть бути обмеження або особливі потреби. Контролі за дизайном та розташуванням елементів повинні враховувати стандарти ергономіки та дизайну інтерфейсу для створення збалансованого та ефективного інструменту генерації QR-кодів в середовищі Unity.

### 3.2.2 Створення програмного коду генератора QR-коду

Розробка програми генератора QR-кодів - це процес створення програмного коду, що взаємодіє з бібліотеками для генерації QR-кодів та керує їх функціоналом. Починаючи з імпорту необхідних бібліотек або плагінів для генерації QR-кодів в середовищі розробки, першим кроком є створення основи програми, яка включає в себе структуру проекту та початкові файли коду.

Наступним етапом є створення логіки програми для обробки введених даних від користувача або іншого джерела. Це може включати отримання текстового вводу, URL-адреси, контактної інформації або інших даних, які будуть використовуватися для створення QR-коду.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using ZXing;
5  using ZXing.QrCode;
6  using UnityEngine.UI;
7  using TMPro;
8
9  public class QRgenerator : MonoBehaviour
10 {
11
12     [SerializeField]
13     private RawImage _rawImageReceiver;
14     [SerializeField]
15     private TMP_InputField _textInputField;
16     private Texture2D _storeEncodedTexture;
17
18     // Start is called before the first frame update
19     void Start()
20     {
21         _storeEncodedTexture = new Texture2D(256, 256);
22     }
```

Рисунок 3.5 – Розробка поведінки програми

```

23
24 private Color32[] Encode(string textForEncoding, int width, int height)
25 {
26     BarcodeWriter writer = new BarcodeWriter
27     {
28         Format = BarcodeFormat.QR_CODE,
29         Options = new QrCodeEncodingOptions
30         {
31             Height = height,
32             Width = width
33         }
34     };
35     return writer.Write(textForEncoding);
36 }
37 public void OnClickEncode()
38 {
39     EncodeTextToQrCode();
40 }
41 private void EncodeTextToQrCode()
42 {
43     string textWrite = string.IsNullOrEmpty(_textInputField.text)
44         ? "You should write something" : _textInputField.text;
45     Color32[] _convertPixelTotexture =
46         Encode(textWrite, _storeEncodedTexture.width, _storeEncodedTexture.height);
47     _storeEncodedTexture.SetPixels32(_convertPixelTotexture);
48     _storeEncodedTexture.Apply();
49     _rawImageReceiver.texture = _storeEncodedTexture;
50 }
51 }

```

Рисунок 3.6 – Розробка поведінки програми

Після створення QR-коду, що представлено на рис. 3.5-3.6 – програма повинна забезпечити його відображення для користувача. Це здійснено шляхом генерації графічного зображення з кодом або використанням вбудованих можливостей інтерфейсу Unity для відображення QR-коду.

Для забезпечення більшої функціональності можуть бути додані додаткові можливості, такі як налаштування розміру, кольорів, рівня корекції помилок та інших параметрів QR-коду, щоб надати користувачеві більше контролю над його виглядом та функціональністю.

Наприкінці розробки важливо перевірити код на відповідність стандартам та правильність роботи у різних сценаріях використання. Крім того, документація програмного коду і створення коментарів може значно полегшити розуміння та подальше редагування коду для інших розробників.



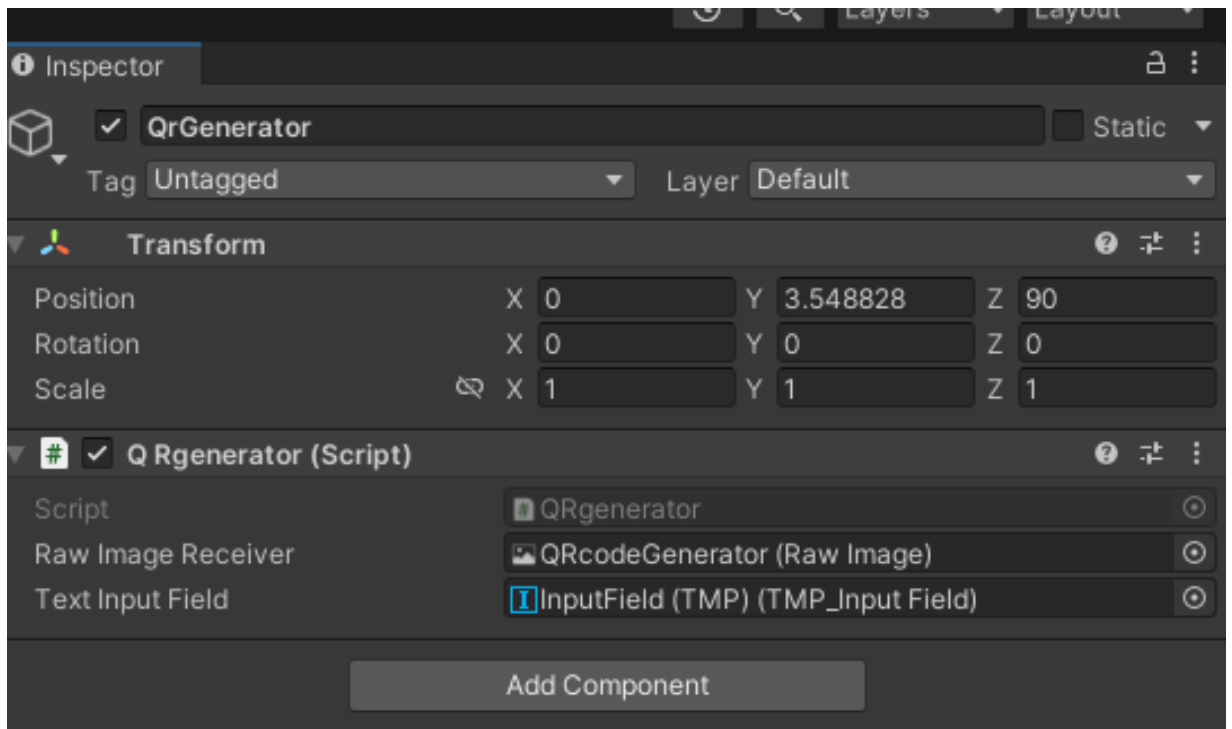


Рисунок 3.7 – Додаємо відповідні компоненти до сценарію

Після прикріплення потрібних компонентів інтерфейсу до розробленого коду, зображено на рис. 3.7 – для коректної роботи інтерфейсу та сценарію програми.

"Canvas" є основним контейнером для UI-елементів, які відображаються на екрані користувача, забезпечуючи створення інтерфейсу гри.

"Image" використовується для відображення графічних елементів у UI, таких як текстури, іконки або зображення.

"Button" та "Text" є компонентами, які дозволяють створювати кнопки та текстові елементи у користувацькому інтерфейсі відповідно.

Ці компоненти інтерфейсу у Unity утворюють основну структуру та функціональність, дозволяючи створювати складні та захоплюючі візуальні ефекти та інтерактивний досвід для користувачів.

Також було додано подію на натискання кнопки, рис. 3.8 – для конвертації набраного тексту в QR-код.

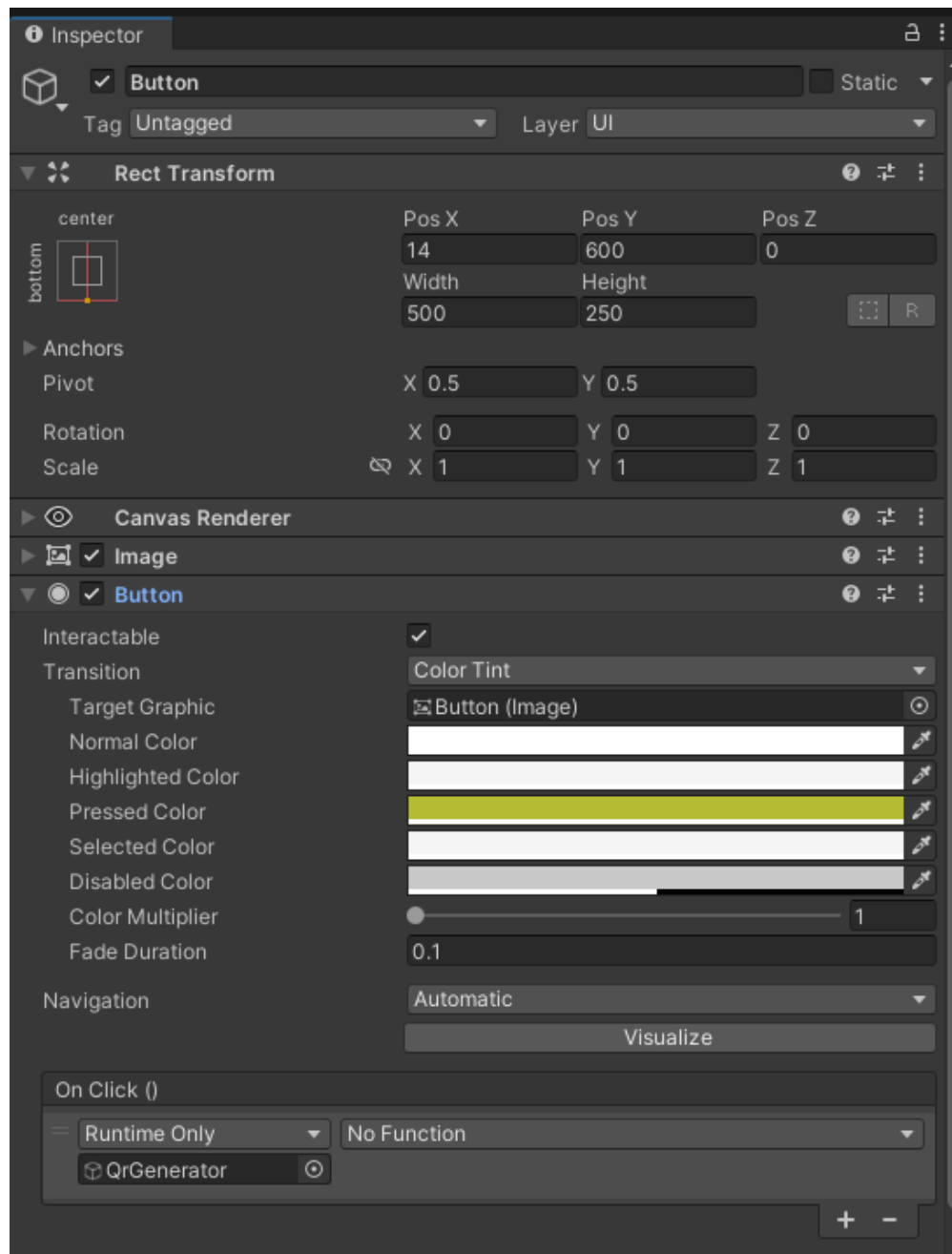


Рисунок 3.8 – Додаємо подію на натискання кнопки

Перевірка роботи програми є критично важливим етапом в процесі розробки для забезпечення її правильної функціональності та відповідності очікуванням користувачів.

Перевірка програми починається з функціональних тестів, що перевіряють правильність роботи окремих функцій або функціональних можливостей програми. Тестування наведено на рис. 3.9.

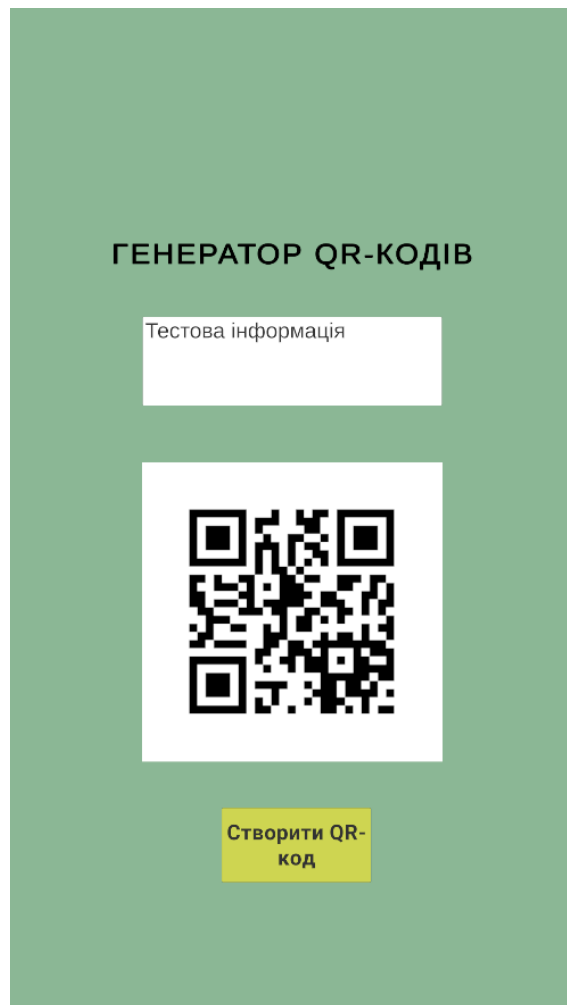


Рисунок 3.9 – Перевіряємо роботу програми

Системні тести виконуються для перевірки програми в цілому. Це випробування всієї функціональності програми в реальних або симульованих умовах використання, щоб переконатися, що вона працює вірно та задовольняє всі вимоги та очікування користувача.

Важливо проводити тестування на різних платформах, пристроях та у різних умовах використання для виявлення можливих проблем, пов'язаних з сумісністю чи продуктивністю програми.

Загалом, ефективна перевірка програми забезпечує надійність, функціональність та задоволення потреб користувача, що робить її ключовим етапом в процесі розробки програмного забезпечення.

### 3.3 Розробка сканера QR-коду

Розробка сканера QR-кодів в середовищі Unity - це завдання, що передбачає створення програмного коду для розпізнавання та інтерпретації інформації, закодованої у вигляді QR-коду за допомогою камери або зображення. Цей процес зазвичай включає в себе використання бібліотек або плагінів, які дозволяють Unity взаємодіяти з камерою та проводити розпізнавання кодів.

Спочатку потрібно імпортувати або встановити відповідний плагін для роботи з камерою в Unity, рис. 3.10. Це може бути здійснено за допомогою спеціальних плагінів або API, які надають доступ до камери пристрою.

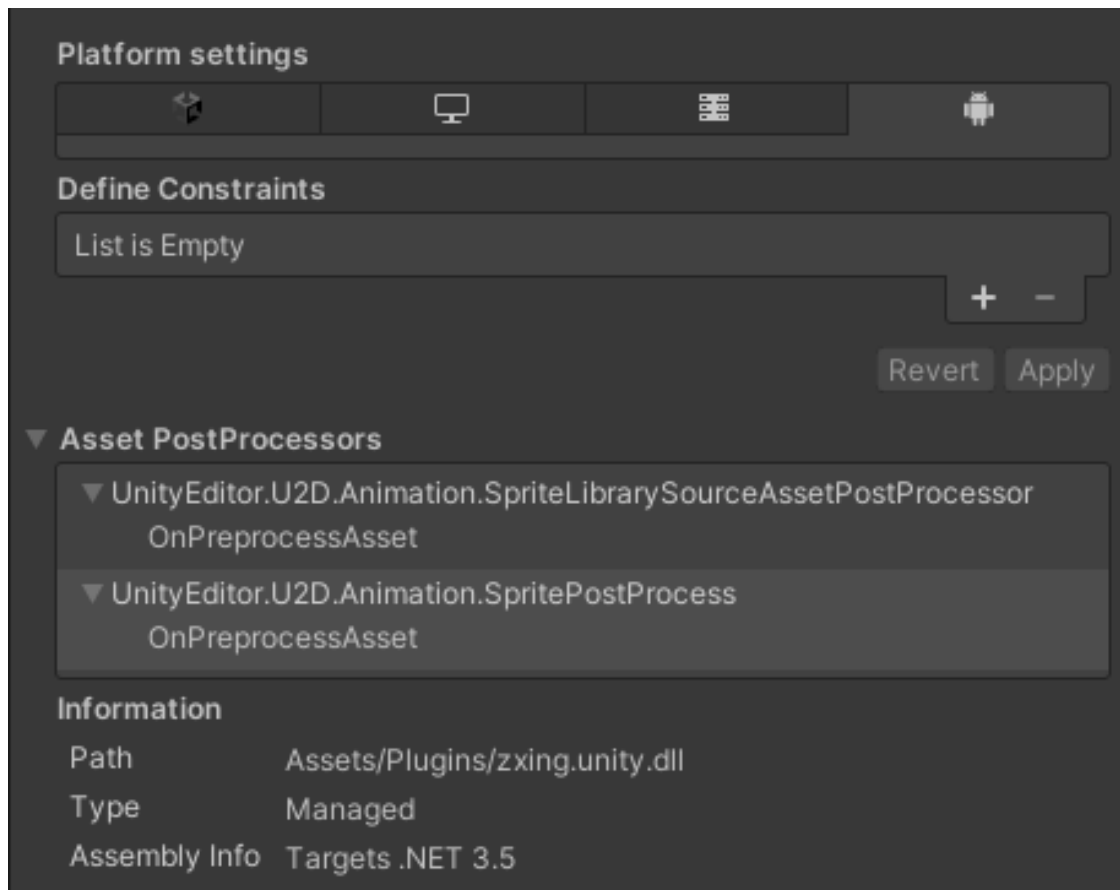


Рисунок 3.10 – Встановлення бібліотеки

### 3.3.1 Розробка інтерфейсу сканера QR-коду

Розробка інтерфейсу сканера QR-кодів у середовищі Unity полягає у створенні елементів користувацького інтерфейсу, що забезпечують зручну та ефективну взаємодію користувача з функціоналом сканування QR-кодів. Це включає в себе створення візуальних компонентів, які дозволяють активувати камеру, показувати зображення з неї та обробляти інформацію, отриману з QR-кодів.

Початковий етап розробки інтерфейсу полягає у визначенні візуальних елементів, таких як кнопки, текстові поля та екрани, що забезпечують зручне керування та відображення роботи сканера QR-кодів. Дизайн інтерфейсу має бути простим, естетичним та забезпечувати зрозумілість функцій користувачам.

Необхідно додати кнопку або спеціальний елемент для активації камери, який відповідає за початок процесу сканування QR-кодів. Також важливо врахувати елементи для відображення зображення з камери та для відображення результатів сканування.

Рекомендовано забезпечити інформативні повідомлення та статуси, що дозволяють користувачу розуміти, коли камера активована, коли вона зчитує QR-код та коли інформація з нього була успішно отримана.

Важливим елементом є реалізація алгоритмів обробки зображення для виявлення та розпізнавання QR-кодів. Це може включати в себе елементи візуалізації діаграм або показників, які демонструють процес розпізнавання.

Потрібно враховувати вимоги до мобільності та різні розміри екранів для створення адаптивного інтерфейсу рис. 3.11 – розробка інтерфейсу, який зручно працює на різних пристроях.

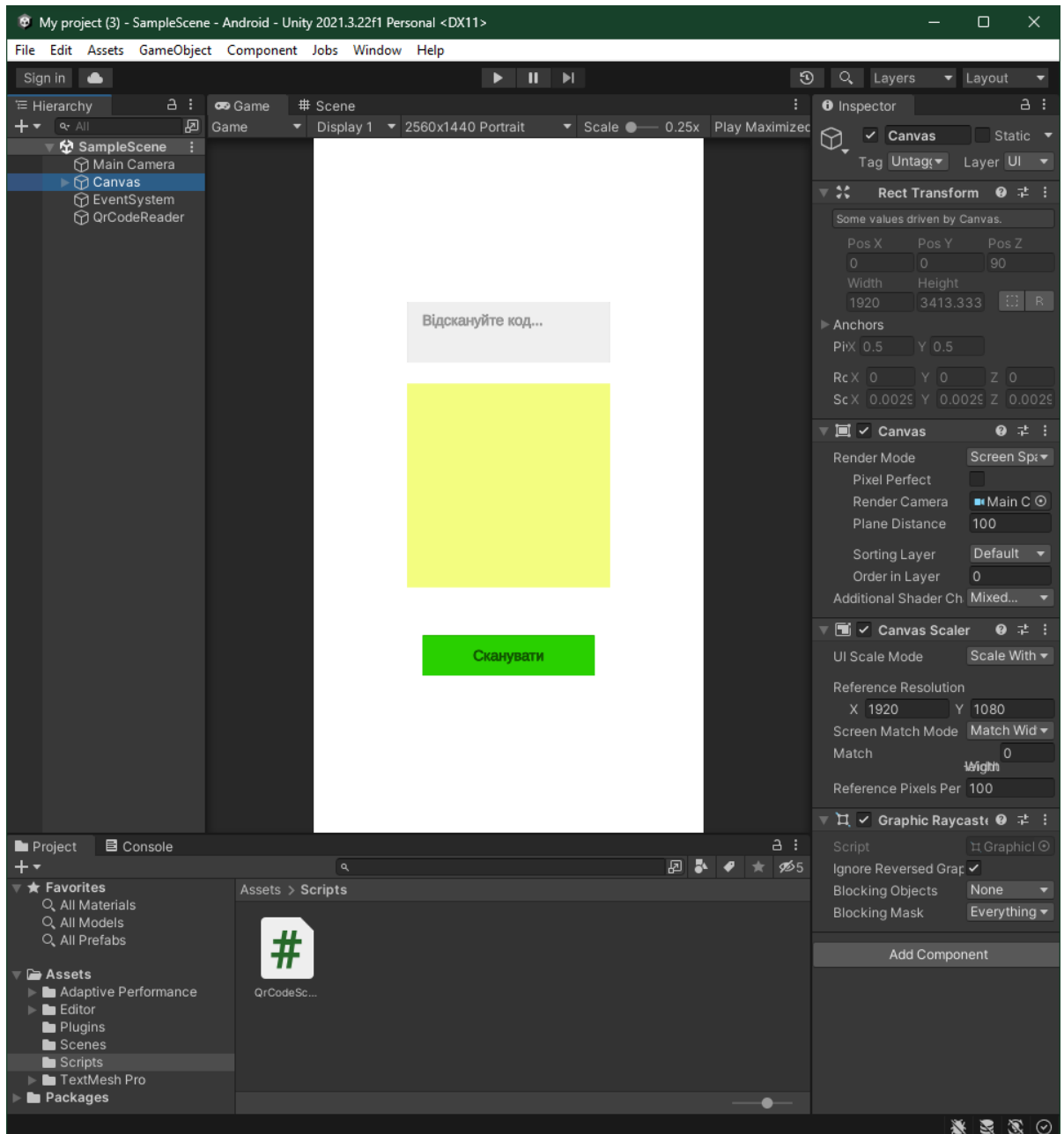


Рисунок 3.11 – Розробка інтерфейсу

### 3.3.2 Розробка програмного коду сканера QR-коду

Далі, необхідно реалізувати логіку, яка дозволить Unity використовувати камеру для зчитування зображення з QR-кодом. Це може включати в себе створення інтерфейсу для відображення зображення з камери та здійснення обробки отриманого зображення для виявлення QR-кодів.

Для обробки зображення та пошуку QR-кодів необхідно використовувати відповідні алгоритми, які забезпечують аналіз зображення та визначають наявність QR-коду. Ці алгоритми можуть базуватися на комп'ютерному зорі, обробці зображень або використовувати спеціалізовані бібліотеки для розпізнавання кодів.

Після виявлення QR-коду, програма повинна витягти закодовану інформацію та обробити її відповідно до необхідних вимог. Це може включати розкодування тексту, URL-адрес, географічних координат або інших типів інформації, які містяться у QR-коді.

Потім, отриману інформацію можна використовувати для подальшої обробки або взаємодії з іншими частинами програми, що залежить від конкретних вимог проекту.

Розробка програмного коду сканера QR-кодів в середовищі Unity вимагає створення алгоритмів обробки зображення та інтерфейсу для взаємодії з камерою пристрою. Цей процес починається з імпортування або використання спеціалізованих плагінів, що надають доступ до камери, а також бібліотек для розпізнавання QR-кодів.

Основний аспект кодування сканера QR-кодів - це інтеграція функціоналу доступу до камери в програму на Unity, рис. 3.12. Це може бути здійснене через виклик API плагіну камери, що дозволяє отримувати зображення з камери для подальшої обробки.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using ZXing;
5  using TMPro;
6  using UnityEngine.UI;
7
8  public class QrCodeScanner : MonoBehaviour
9  {
10     [SerializeField]
11     private RawImage _rawImageBackground;
12     [SerializeField]
13     private AspectRatioFitter _aspectRatioFitter;
14     [SerializeField]
15     private TextMeshProUGUI _textOut;
16     [SerializeField]
17     private RectTransform _scanZone;
18
19     private bool _isCamAvaible;
20     private WebCamTexture _cameraTexture;
21
22     // Start is called before the first frame update
23     void Start()
24     {
25         SetupCamera();
26     }
27
28     // Update is called once per frame
29     void Update()
30     {
31         UpdateCameraRender();
32     }
33

```

Рисунок 3.12 – Розробка сканеру

У рамках розроблення програмного забезпечення для візуалізації та обробки відеоданих у середовищі Unity, було створено клас «SetupCamera». Цей клас спрямований на керування камерою пристрою та надання доступу до відеопотоку для подальшої обробки візуальної інформації, рис. 3.13. Клас «SetupCamera» реалізує основні методи для підключення, керування та отримання відеоданих з камери.



```

34 private void SetUpCamera()
35 {
36     WebCamDevice[] devices = WebCamTexture.devices;
37
38     if (devices.Length == 0)
39     {
40         _isCamAvaible = false;
41         return;
42     }
43     for (int i = 0; i < devices.Length; i++)
44     {
45         if (devices[i].isFrontFacing == false)
46         {
47             _cameraTexture = new WebCamTexture(devices[i].name,
48                 (int)_scanZone.rect.width,
49                 (int)_scanZone.rect.height);
50         }
51     }
52
53     _cameraTexture.Play();
54     _rawImageBackground.texture = _cameraTexture;
55     _isCamAvaible = true;
56 }

```

Рисунок 3.13 –Створення класу камери

Клас «SetUpCamera» має методи, спрямовані на ініціалізацію камери, встановлення параметрів зображення (роздільна здатність, частота кадрів тощо), отримання та обробку кадрів в реальному часі, а також завершення роботи з камерою, рис. 3.14.

Крім того, клас має можливість обробки помилок, які можуть виникнути під час підключення чи роботи з камерою, забезпечуючи відповідну обробку виняткових ситуацій.

Під час розробки методів класу «SetUpCamera», враховувалися основні принципи організації та управління ресурсами, що дозволяє ефективно використовувати функціонал камери та мінімізувати навантаження на системні ресурси.

```

59  public void UpdateCameraRender()
60  {
61      if (_isCamAvaible == false)
62      {
63          return;
64      }
65      float ratio = (float)_cameraTexture.width / (float)_cameraTexture.height;
66      _aspectRatioFitter.aspectRatio = ratio;
67
68      int orientation = -_cameraTexture.videoRotationAngle;
69      _rawImageBackground.rectTransform.localEulerAngles = new Vector3(0, 0, orientation);
70
71  }

```

Рисунок 3.14 – Створення класу відображення зображення камерою

Загальна архітектура класу передбачає можливість розширення функціональності для підтримки різноманітних типів камер, оптимізації обробки відеоданих та додаткових можливостей управління камерою для відповідності специфічним потребам проектів у візуальній обробці та аналітиці.

```

73  public void OnClickScan()
74  {
75      Scan();
76  }
77
78  private void Scan()
79  {
80      try
81      {
82          IBarcodeReader barcodeReader = new BarcodeReader();
83          Result result = barcodeReader.Decode(_cameraTexture.GetPixels32(),
84              _cameraTexture.width,
85              _cameraTexture.height);
86          if (result != null)
87          {
88              _textOut.text = result.Text;
89          }
90          else
91          {
92              _textOut.text = "FAILED TO READ QR CODE";
93          }
94      }
95      catch
96      {
97          _textOut.text = "FAILED IN TRY";
98      }
99  }
100 }
101

```

Рисунок 3.15 – Створення класу сканування

Клас «Scan» поєднує в собі основні функції управління камерою та обробки з функціоналом розпізнавання QR-кодів, рис. 3.15. Він ініціює активність камери, забезпечує реальний час зйомки та передачу потоку відеоданих для подальшого аналізу та виявлення QR-кодів у цьому потоці.

Під час створення методів класу «Scan» враховувалася оптимальна організація алгоритмів пошуку та розпізнавання QR-кодів для забезпечення швидкості та точності процесу сканування.

Після виявлення QR-коду, програма має провести його розпізнавання за допомогою спеціалізованих бібліотек для розпізнавання QR-кодів. Це включає в себе використання методів читання та декодування інформації, закодованої в QR-коді.

Звертаючись до структури коду, важливо створити модульну архітектуру, розділяючи функціональність на логічні блоки, що полегшує тестування та підтримку. Документування коду та додавання коментарів є необхідними для полегшення розуміння коду іншими розробниками.

Також важливим аспектом є робота з інтерфейсом, що відображає зображення з камери та результати сканування. Це включає в себе створення візуальних компонентів, таких як екрани, кнопки та повідомлення, які забезпечують зручну взаємодію користувача з програмою.

Узагальнюючи, розробка програмного коду сканера QR-кодів у середовищі Unity - це процес інтеграції функцій доступу до камери, обробки зображень, пошуку та розпізнавання QR-кодів, а також створення інтерфейсу для зручної взаємодії з користувачем.

На рис. 3.16 зображено тестування роботи додатка, зі статусом «Успішно», та на рис. 3.17 зі статусом «Не знайдено код».

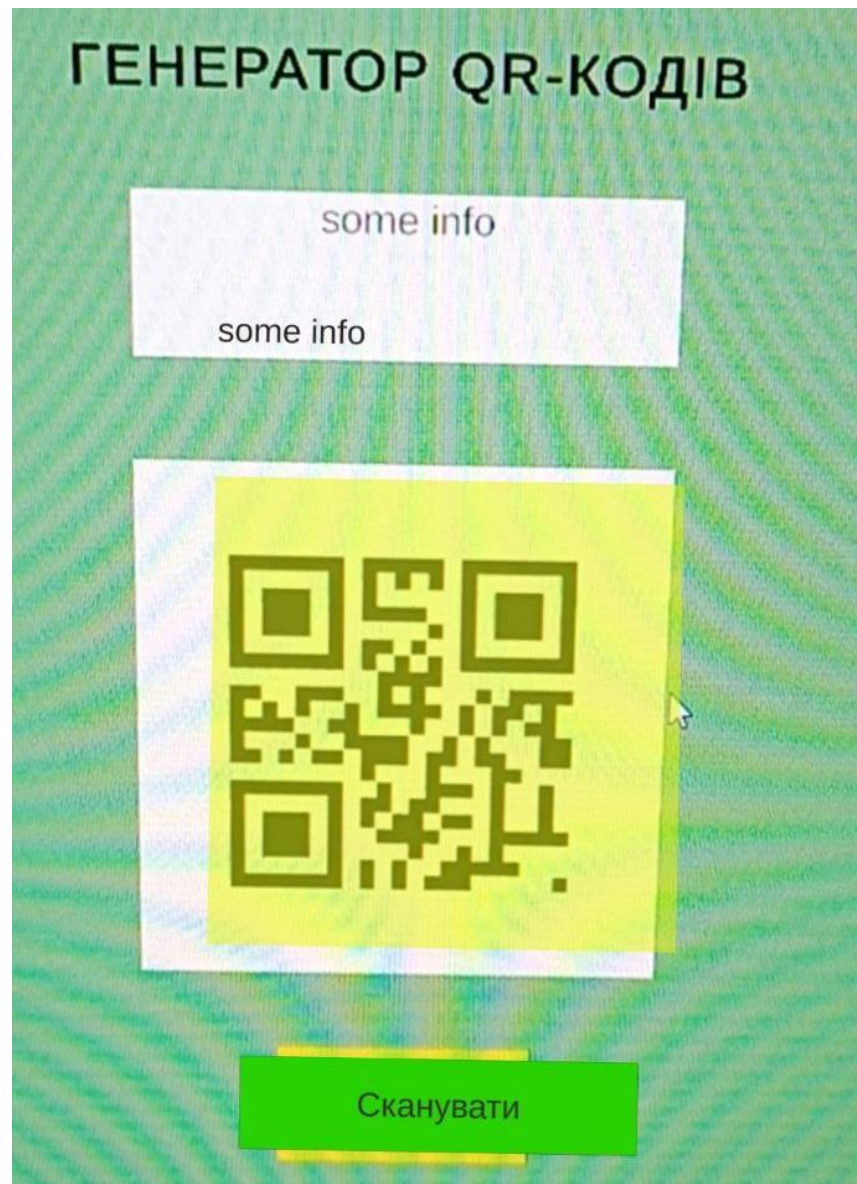


Рисунок 3.16 – Тестування сканера з результатом «Успішно»

Тестування проводилося у контрольованому середовищі з використанням стабільного програмного забезпечення на основі Unity з інтегрованим модулем сканування QR-кодів. Для тестування були використані стандартні QR-коди різних розмірів та складності.

Термін "удача" застосовувався до сканів, які відповідали передбаченим вимогам успішного зчитування: відсутність помилок у розпізнаванні, швидкість та точність зчитування, а також можливість подальшого коректного використання отриманої інформації.

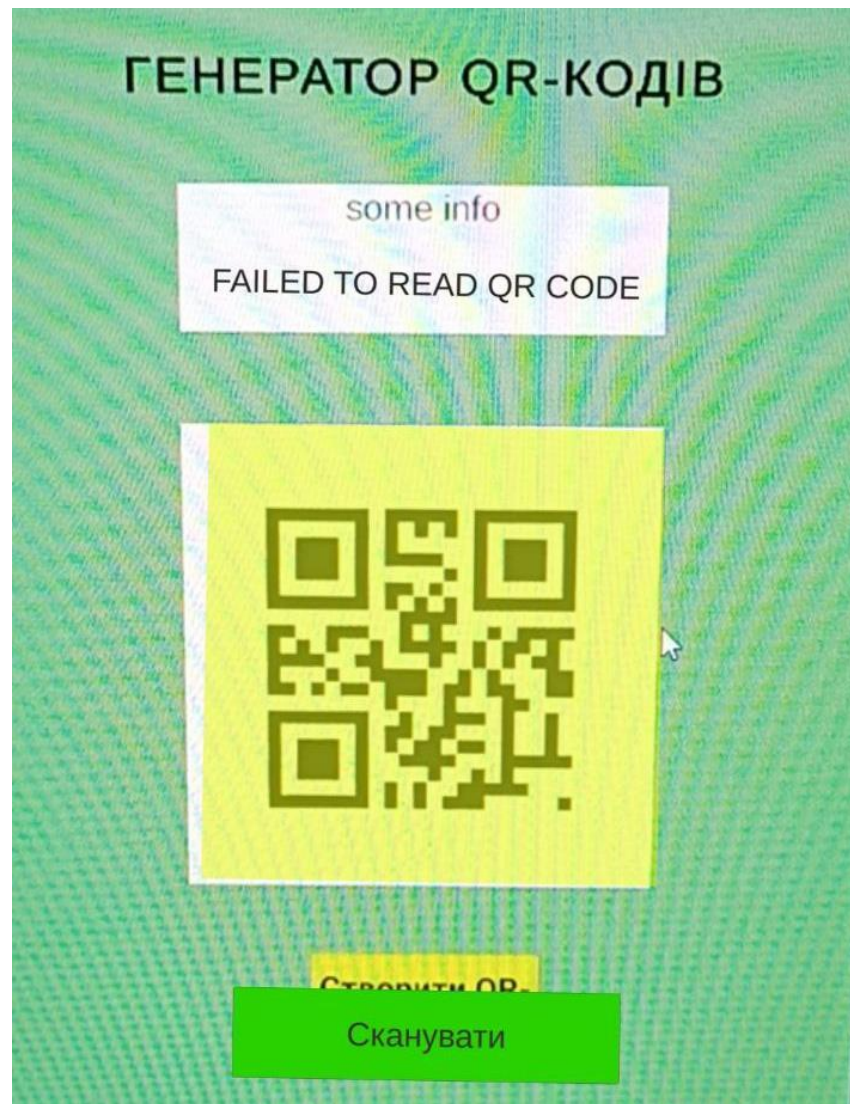


Рисунок 3.17 – Тестування сканера з результатом «Невдача»

Процес тестування сканера QR-кодів зі статусом "невдача" відображає результати, коли сканер не зміг успішно розпізнати чи ідентифікувати QR-коди, не досягнувши очікуваних критеріїв успішності. Цей процес важливий для виявлення обмежень сканера та вдосконалення його алгоритмів розпізнавання.

Невдачею були визначені випадки, коли сканер не зміг зчитати код, спотворив його чи зробив неправильний розпізнавальний висновок. Важливими були також випадки довгого часу розпізнавання або відсутності реакції сканера на QR-коди. На основі отриманих результатів тестування було зроблено висновок про причини невдачі сканера QR-кодів. Цей висновок включав можливі шляхи вдосконалення алгоритмів сканування для покращення точності та швидкості розпізнавання.

### 3.4 Розробка програмного коду сортувального алгоритму

Розробка сортувального алгоритму для автоматизованої системи сортування об'єктів за допомогою QR-кодів представляє собою важливу задачу, яка дозволяє ефективно керувати та відстежувати об'єкти в різних галузях, таких як логістика, наука та промисловість.

Алгоритм призначений для сортування об'єктів за допомогою QR-кодів, які містять у собі інформацію про об'єкт. Це дозволяє ефективно відстежувати та організувати об'єкти в системі.

Важливо ретельно враховувати аспекти оптимізації пам'яті та забезпечення стійкості алгоритму до різних типів вхідних даних. Код алгоритму виконується відповідно до визначених алгоритмічних принципів, з дотриманням структури та стилю коду, щоб забезпечити його зрозумілість та підтримку. У процесі реалізації важливо враховувати можливості паралельності та оптимізації для забезпечення ефективності роботи алгоритму на сучасних обчислювальних системах.

Алгоритм базується на швидкому та ефективному сортуванні за унікальними QR-кодами. Застосовується алгоритм QuickSort, який розділяє об'єкти на підмасиви за унікальними ідентифікаторами QR-кодів та рекурсивно сортує їх. Програмний код алгоритму наведено на рис. 3.18.

```

1  using System;
2  using System.IO;
3
4  class FileAlphabeticalSort
5  {
6      // Метод для сортування рядків за алфавітом
7      static void SortLinesAlphabetically(string filePath, string newFilePath)
8      {
9          try
10         {
11             // Читання усього вмісту файлу
12             string[] lines = File.ReadAllLines(filePath);
13
14             // Сортування рядків за алфавітом
15             Array.Sort(lines, StringComparer.CurrentCulture);
16
17             // Запис відсортованого вмісту у новий файл
18             File.WriteAllLines(newFilePath, lines);
19
20             Console.WriteLine("Файл був відсортований та перезаписаний з новою назвою.");
21         }
22         catch (Exception e)
23         {
24             Console.WriteLine("Виникла помилка: " + e.Message);
25         }
26     }
27
28     static void Main()
29     {
30         string path = "test.txt";
31         string newPath = "sortedTest.txt";
32
33         // Виклик методу сортування рядків у файлі за алфавітом з перезаписом у новий файл
34         SortLinesAlphabetically(path, newPath);
35     }
36 }
37

```

Рисунок 3.18 – Програмний код алгоритму QuickSort

Розробка алгоритму сортування об'єктів за допомогою QR-кодів вимагає ретельного випробування для впевненості в його ефективності та надійності.

Файл з вхідними даними, з ідентифікатором напрямки(міста) зображено на рис. 3.19.

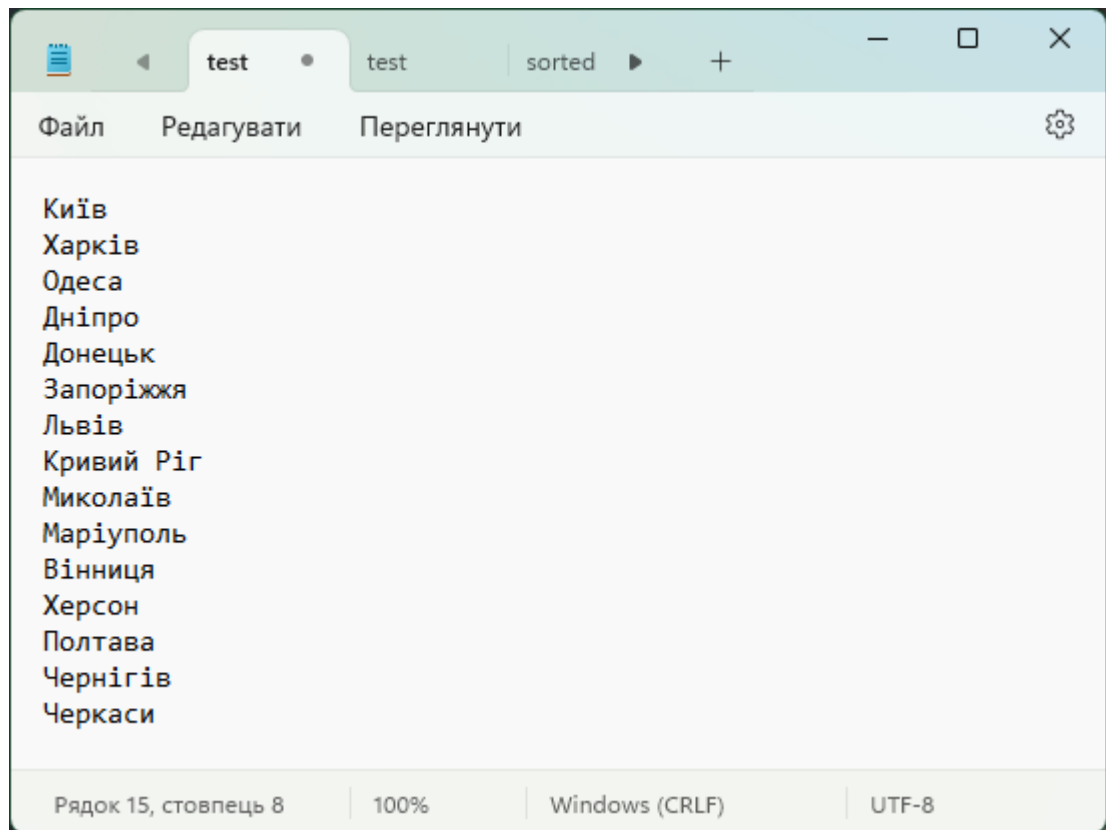


Рисунок 3.19 – Вхідні дані

Процес випробування має на меті виявлення можливих проблем, оптимізацію продуктивності та забезпечення високої якості роботи системи. Важливо враховувати різноманітні аспекти, такі як швидкість обробки даних, точність сортування, обробка помилок і реалізація безпеки.

Випробування були розпочаті створенням тестового набору об'єктів, кожен з яких був обладнаний унікальним QR-кодом, та представлений ідентифікатором об'єкта. Об'єкти були систематизовані та розміщені у випадковому порядку.

Алгоритм сортування був застосований до тестового набору з вимірюванням часу сортування та оцінкою правильності порядку об'єктів після сортування, зображено на рис. 3.20. Також проводилося вивчення продуктивності при обробці різної кількості об'єктів.



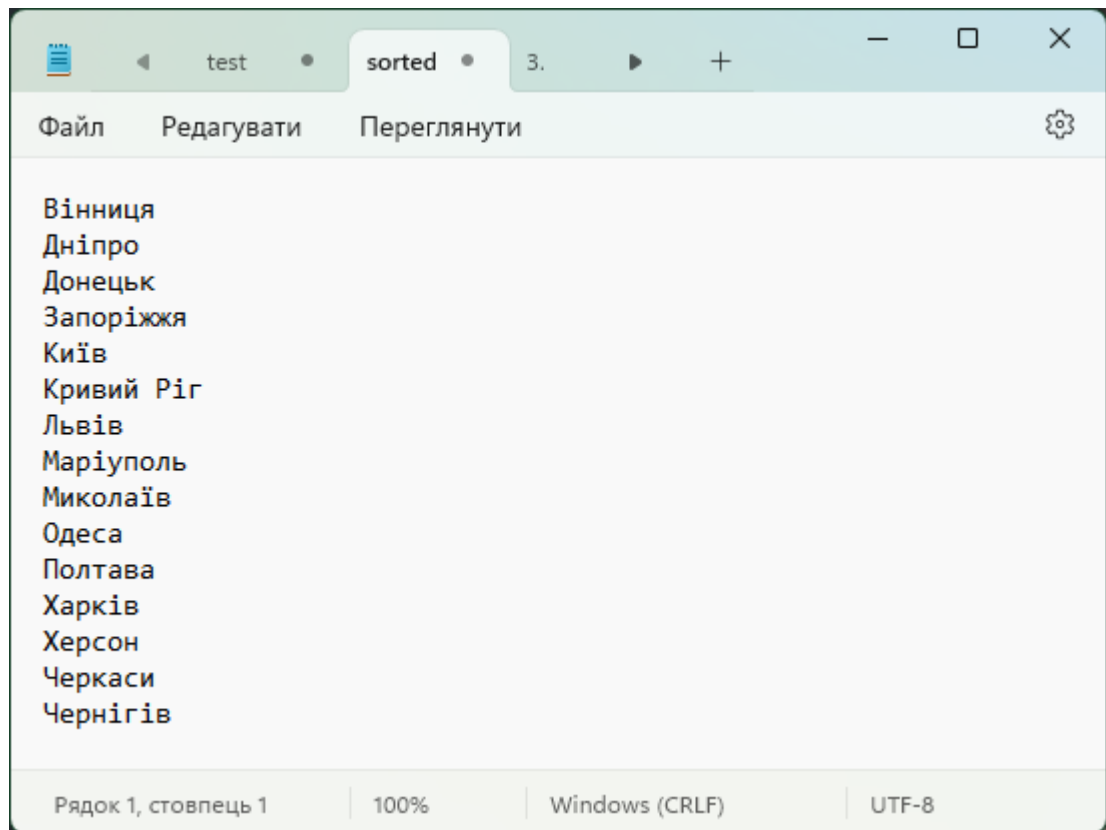


Рисунок 3.20 – Вихідні дані

## Висновки до розділу

Розробка сканера QR-кодів у середовищі Unity - це складний та важливий процес, що вимагає інтеграції функціоналу доступу до камери, алгоритмів обробки зображень та функцій розпізнавання QR-кодів для створення ефективного інструменту сканування. Це забезпечує зручність інтерфейсу користувача та реалізацію функціональності сканування QR-кодів в реальному часі.

Важливою частиною розробки є імпорт або використання плагінів для камери та визначення методів отримання зображення з неї для подальшої обробки. Це надає можливість отримувати вхідні дані для аналізу та розпізнавання QR-кодів.

Алгоритми обробки зображень використовуються для пошуку QR-кодів, виявлення їх на зображеннях та виділення областей з кодами для подальшого

аналізу. Це може включати в себе роботу з контурами, пікселями та визначенням характерних ознак QR-кодів.

Після виявлення QR-коду, використання спеціалізованих бібліотек дозволяє декодувати та отримувати інформацію, закодовану в QR-коді. Це важливо для отримання коректних та розшифрованих даних з розпізнаного QR-коду.

На основі зібраних даних та аналізу результатів тестування було зроблено висновок про ефективність сканера QR-кодів, підтверджуючи або відхиляючи його статус "удача" залежно від відповідності встановленим вимогам та критеріям успішності сканування.

Створення ефективного інтерфейсу включає в себе визначення та реалізацію візуальних компонентів, які дозволяють користувачам активувати камеру, спостерігати за зображенням з неї та бачити результати сканування.

Архітектура програмного забезпечення розрахована на можливість розширення функціональності для підтримки різних типів QR-кодів, оптимізації алгоритмів розпізнавання та внесення модифікацій для відповідності специфічним вимогам проектів у сфері сканування та обробки QR-кодів.

Узагальнюючи, розробка сканера QR-кодів у Unity вимагає глибоких знань алгоритмів обробки зображень, використання бібліотек для розпізнавання QR-кодів та навичок роботи з інтерфейсами для створення функціонального та ефективного інструменту сканування, що забезпечує зручну взаємодію з користувачем.

## 4 ТЕСТУВАННЯ ТА ДОСЛІДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Тестування генератора QR-коду

Метою даного дослідження є аналіз ефективності програмного забезпечення, призначеного для створення QR-кодів, що є широко використовуваними для зберігання інформації та передачі даних. В рамках цього експерименту проводиться систематичне тестування різних аспектів функціонування програми з метою визначення її надійності та продуктивності. Методика тестування передбачає введення різноманітних вхідних даних, від великих обсягів тексту до спеціальних символів, для перевірки коректності та швидкості генерації QR-кодів.

Застосовується статистичний підхід до аналізу результатів, що дозволяє виявити можливі відхилення в роботі програми та їхні причини. В ході дослідження ретельно вивчаються параметри кодування, розмір і склад складних даних, що впливає на стійкість та якість отриманих QR-кодів. Для оцінки адекватності програми також проводиться порівняння її результатів із стандартами та вимогами до QR-кодів, що прийняті в індустрії. Результати експерименту аналізуються з використанням спеціалізованих програмних інструментів та математичних моделей для об'єктивної оцінки точності та стабільності роботи програмного продукту.



Рисунок 4.1 – Тестування генератора

Отримані дані піддаються статистичному аналізу з метою визначення середніх значень, стандартних відхилень та інших параметрів, що визначають характеристики роботи програми. Тестування зображено на рис. 4.1. – 4.2. Враховуючи знайдені закономірності та результати, формуються рекомендації щодо оптимізації роботи програми для підвищення її ефективності та стійкості у виробничому середовищі. Такий підхід дозволяє провести комплексний аналіз функціональності програмного забезпечення для генерації QR-кодів та забезпечити його високу якість та надійність у реальних умовах застосування.

Також було проведено тестування в додатку «Google Об'єктив» для перевірки та незалежної оцінки роботи розробленого генератора QR-кодів, що зображено на рис. 4.3.



а)

б)

Рисунок 4.2 – Тестування генератора

Тестування генератора в додатку «Google Об'єктив» спрямоване на перевірку функціональності генератора в контексті його основних можливостей, зокрема, розпізнавання тексту на фотографіях, здатність вибору мови та точність розпізнавання різних шрифтів та форматів.

Тестування включає в себе аналіз точності та якості розпізнавання тексту на різних типах зображень - чітких, розмитих, з різним освітленням та кутами огляду, щоб оцінити стійкість генератора до візуальних аномалій.

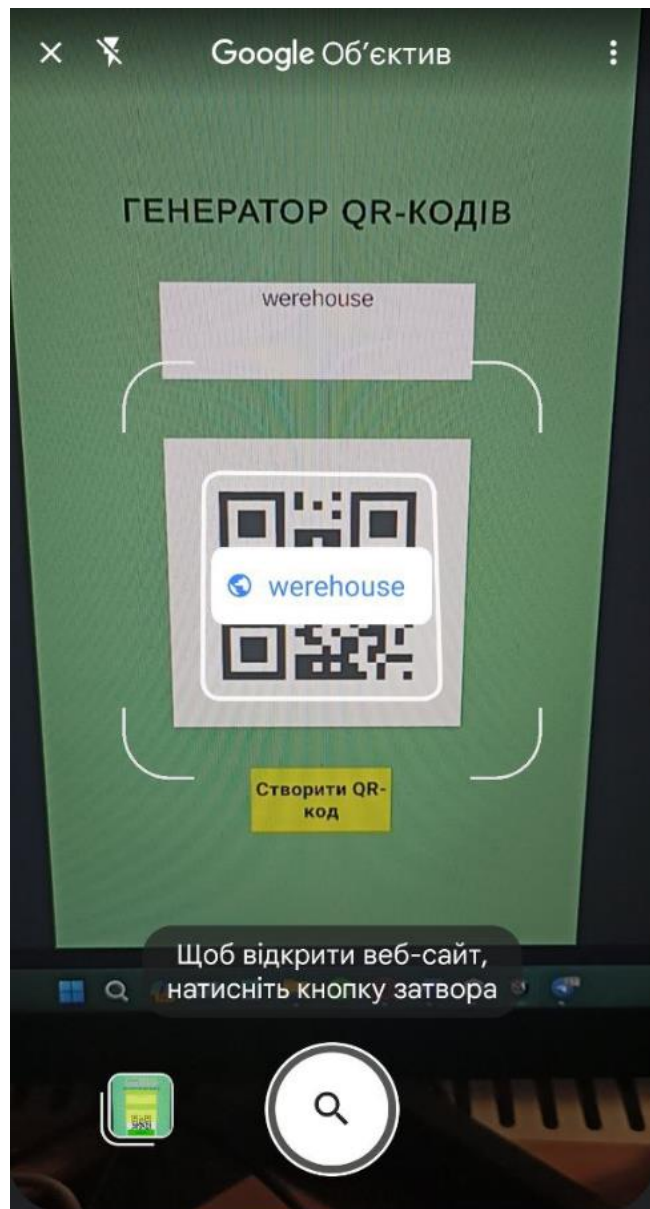


Рисунок 4.3 – Тестування генератора в додатку «Google Об'єктив»

Оцінка часу, який потрібно генератору для розпізнавання та конвертації тексту на зображеннях, для визначення ефективності та швидкодії інструменту.

Проведення тестів для оцінки стійкості роботи генератора при великому обсязі робочих завдань, щоб визначити його стабільність та ефективність під час інтенсивного використання.

Після завершення тестів аналізуються отримані результати для визначення ефективності та точності роботи генератора.

## 4.2 Тестування сканера QR-коду

Дане дослідження націлене на аналіз функціональності програмного забезпечення, спрямованого на сканування QR-кодів, які використовуються для зчитування та інтерпретації інформації. Метою експерименту є систематичне тестування різних аспектів роботи програми з метою визначення її точності, швидкодії та функціональності. Методика дослідження включає в себе аналіз реакції програми на різноманітні типи QR-кодів, від стандартних до складних варіантів, з різними рівнями складності та розмірів. Проводиться тестування з використанням різних пристроїв та операційних систем з метою визначення універсальності та сумісності програмного забезпечення. Експеримент включає в себе оцінку швидкості сканування, розпізнавання та інтерпретації QR-кодів з різних джерел та у різних умовах освітлення.

Ключовою частиною дослідження є аналіз точності та стійкості програми до помилок, зокрема у випадках, коли QR-коди пошкоджені, частково затемнені або розмиті. Для цього створюються спеціальні тестові набори з різними видами пошкоджень QR-кодів для вивчення поведінки програми в таких ситуаціях. Важливим етапом експерименту є також аналіз здатності програми працювати з різними типами даних, закодованих у QR-кодах, включаючи текст, URL-адреси, контактну інформацію, номери телефонів та інші формати.

Тестування програми зображено на рис. 4.4 – 4.5.

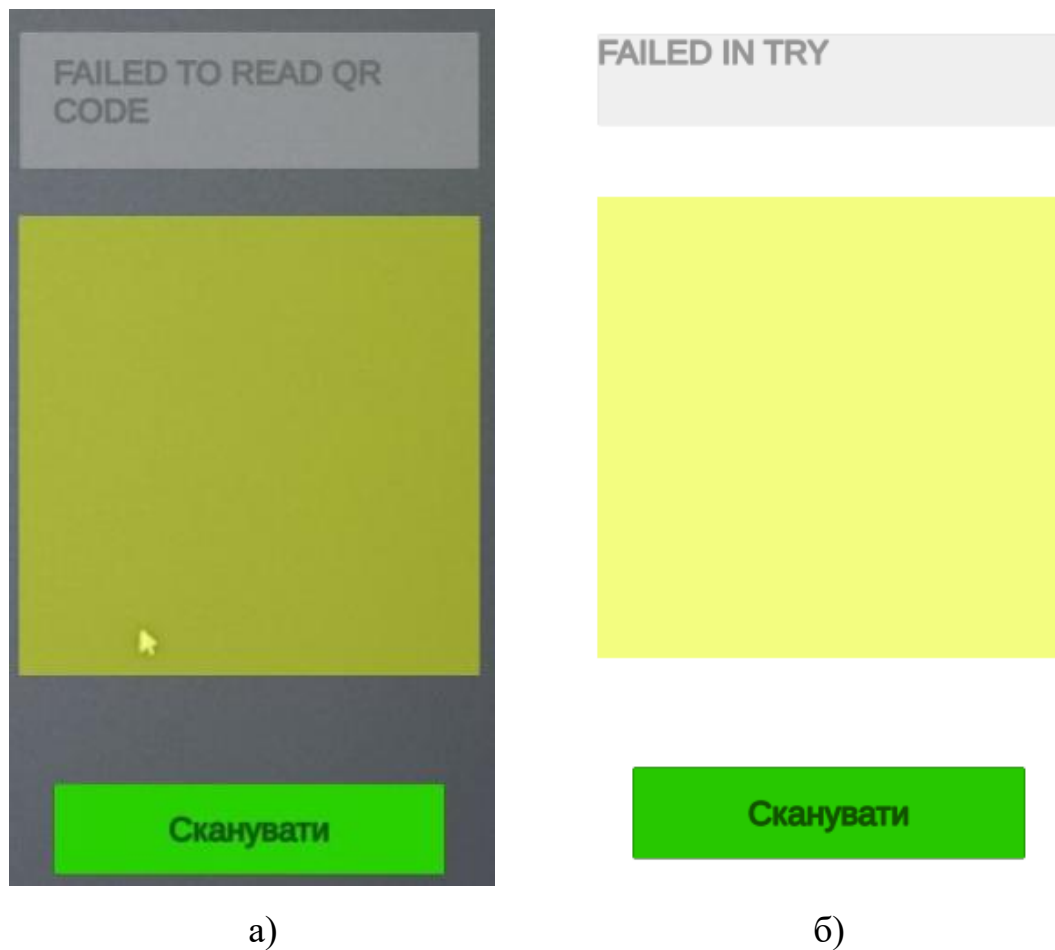


Рисунок 4.4 – Тестування сканера зі статусом «Невдача»

Сканер може виявляти "невдачу" у різних сценаріях, коли не вдається визначити або розпізнати QR-коди чи інші елементи.

Неправильний або пошкоджений QR-код: Якщо код пошкоджений, надрукований не чітко або не відповідає стандартам, сканер може виявити "невдачу" у його розпізнаванні.

Якщо на QR-коді є якісь перешкоди, такі як маркери, додатковий шум або інші об'єкти, сканер може не зможти точно його розпізнати.

Ці ситуації викликають "невдачу" сканера у розпізнаванні QR-кодів, що потребує подальшого вдосконалення та оптимізації алгоритмів сканування для покращення точності та стійкості у різних умовах.





а)

б)

Рисунок 4.5 – Тестування сканера зі статусом «Вдало»

Отримані результати тестування піддаються статистичному аналізу з метою визначення середніх часів сканування, частоти помилок та точності розпізнавання інформації з QR-кодів. Дослідження також враховує вплив різних параметрів, таких як роздільна здатність камери, на результати сканування. На основі аналізу результатів формуються рекомендації з покращення якості та продуктивності програми для оптимального функціонування в різноманітних умовах експлуатації.

Ця методологія дозволяє здійснити комплексний аналіз програмного забезпечення для сканування QR-кодів, забезпечуючи його надійність та ефективність у реальних умовах використання та сприяючи подальшому розвитку цієї технології.

## Висновки до розділу

Після ретельного аналізу функціональності програм для генерації QR-кодів та їхнього подальшого сканування вдалося виявити ключові аспекти їхньої роботи. Дослідження генераторів QR-кодів показало, що програми ефективно працюють з різноманітними типами даних та параметрами кодування, однак виявлено деякі обмеження щодо оптимальних режимів роботи при обробці великих обсягів інформації. Оцінка точності та стійкості сканерів QR-кодів показала, що програмне забезпечення демонструє високу працездатність у розпізнаванні різноманітних кодів, включаючи пошкоджені та складні. Однак було виявлено, що при певних умовах освітлення та спотворень деякі сканери можуть показувати низьку ефективність.

Отримані результати дослідження свідчать про потенціал подальшого вдосконалення програмного забезпечення для створення QR-кодів у напрямку оптимізації роботи з об'ємними даними та підвищення їхньої стійкості до різноманітних параметрів кодування. Щодо сканерів QR-кодів, виявлені обмеження дають можливість вдосконалити алгоритми розпізнавання для підвищення точності у складних умовах експлуатації. Це може стати основою для подальшого розвитку та удосконалення технології QR-кодів у сферах, де вимагається висока надійність та швидкодія системи зчитування. Такі висновки вказують на необхідність подальших досліджень та розробок для досягнення оптимальної роботи програмного забезпечення QR-кодів у різноманітних сценаріях використання.

## ВИСНОВКИ

В результаті виконання магістерської роботи було проаналізовано методи сортування об'єктів на конвеєрі, розроблено структуру системи, її програмне забезпечення. Було проведено тестування та дослідження програмного забезпечення.

Під час виконання роботи було розроблено та досліджено програмне забезпечення, а саме – алгоритми сортування, програма для генерації QR-кодів та сканер для читання QR-кодів. Було виявлено, що головними перевагами є швидкість зчитування, в своїй структурі QR-коди мають вбудованій структурі і коректуючі коди, можливість містити в одному коді багато інформації.

Метою дослідження, було технологічне оновлення системи сортування об'єктів за QR-кодом на базі рушія Unity, та оптимізація робочих процесів використовуючи алгоритм швидкого сортування. Дослідження було спрямоване на вдосконалення алгоритмів обробки даних, що дозволяють забезпечити швидке та точне визначення інформації з QR-кодів, а також виконання точної ідентифікації та класифікації об'єктів на основі цих даних.

Результати дослідження можуть бути використані для подальшого розвитку технологій сортування, що використовують QR-коди. Шляхом вдосконалення алгоритмів обробки даних та розширення можливостей інтеграції з іншими технологіями, можливо досягти ще вищого рівня ефективності та автоматизації в різних галузях.

Загалом, використання автоматизованих систем сортування за QR-кодами є перспективним та дієвим рішенням для підприємств, що мають зацікавленість у підвищенні ефективності сортувальних та виробничих процесів, оптимізації обслуговування та зниженні витрат.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. A review on internet of things (IoT) [Електронний ресурс] – URL: [https://www.researchgate.net/profile/Nirmala-Svsg/post/What are the latest developments in IOT architectures](https://www.researchgate.net/profile/Nirmala-Svsg/post/What%20are%20the%20latest%20developments%20in%20IOT%20architectures) (Дата звернення 10.09.2023)
2. QR-код [Електронний ресурс] – URL: <https://uk.wikipedia.org/wiki/QR-код> (Дата звернення 10.10.2023)
3. Isiramen, Oseikhuemen E. Design and implementation of an automatic conveyor sorting system. Diss. Murdoch University, 2018.
4. Automatic Identification and Data Capture [Електронний ресурс] //SearchGoogle. Електрон. текстові дані. – Режим доступу: – URL: <https://apnews.com/61904f62798e4065a041dc9f17759ea4/> (Дата звернення 15.10.2023).
5. Комплекс автоматизованого керування конвеєрами [Електронний ресурс]. – URL: <https://uk.wikipedia.org> (Дата звернення 10.10.2023)
6. Стрічковий конвеєр [Електронний ресурс] – URL: <https://kapelou.com/blog/intralogistika/strichkovij-konveier-riznovidi-osoblivosti/> (Дата звернення 12.10.2023)
7. RFID History [Електронний ресурс] – URL: <https://www.electronicnotes.com/articles/connectivity/rfid-radio-frequency-identification/developmenthistory.php> (Дата звернення 13.10.2023)
8. Huang, Mengyu, et al. "Optimal Design of a Conveyor-Based Automatic Sorting System." 2020 IEEE 16th International Conference on Control & Automation (ICCA). IEEE, 2020.
9. Caramida, Mihai Stefan. "Automatic Sorting System." Journal of Electrical Engineering, Electronics, Control and Computer Science 2.3 (2017): 5-10.
10. Kottalil, Ancy M., et al. "Automatic sorting machine." Journal for Research 2.04 (2016).

11. Dabade, Sanjay Prakash, and Rohan Prakash Chumble. "Automatic Sorting Machine Using Conveyor Belt." *International Journal of Innovative and Emerging Research in Engineering* 2.5 (2015): 66-70.
12. Object Detection for Smart Factory Processes by Machine Learning. *Procedia Computer Science*. / Lukas Malburg, Manfred-Peter, Seiger Ronny, Klein Patrick, Bergmann Ralph. – 2021p.
13. Malburg L., Klein P., Bergmann R., Semantic Web Services for AI-Research with Physical Factory Simulation Models in Industry 4.0, in: *Proc. of the Int. Conf. on Innov. Intell. Ind. Prod. and Logis. (IN4PL)*. – 2020p.
14. Seredkin, A. V., et al. "Development of a method of detection and classification of waste objects on a conveyor for a robotic sorting system." *Journal of Physics: Conference Series*. Vol. 1359. No. 1. IOP Publishing, 2019.
15. Huang, Mengyu, et al. "Optimal Design of a Conveyor-Based Automatic Sorting System." *2020 IEEE 16th International Conference on Control & Automation (ICCA)*. IEEE, 2020.
16. Deshmukh, Amir, Mahesh Nagane, and Vaibhav Awatade. "Design & Development of Automatic Conveyor Controlling System for Sorting of Component on Color basis." *Red* 630 (2016): 780.
17. OpenCV [Электронный ресурс] – URL:<https://uk.wikipedia.org/wiki/OpenCV>
18. Billingham M. *Emerging Technologies of C#*. / Billingham Mark, Thomas Bruce, 2007. - p. 209 - ISBN 978-1-5990-4066-0.
19. Tho, Tuong Phuoc, Nguyen Truong Thinh, and Nguyen Huy Bich. "Design and development of the vision sorting system." *2016 3rd International Conference on Green Technology and Sustainable Development (GTSD)*. IEEE, 2016.
20. Tripathi, Sameer, et al. "Smart industrial packaging and sorting system." *Strategic system assurance and business analytics* (2020): 245-254.
21. Liang, Kun, et al. "Development and parameter optimization of automatic separation and identification equipment for grain tracing systems based on grain tracers with QR codes." *Computers and Electronics in Agriculture* 162 (2019): 709-718.

- 22.Huang, Mengtao, and Yifan Li. "Express sorting system based on two-dimensional code recognition." 2018 International Conference on Sensor Networks and Signal Processing (SNSP). IEEE, 2018.
- 23.Unity [Электронный ресурс] – URL: <https://unity.com> (Дата звернення 16.11.2023)
- 24.Zxing [Электронный ресурс] – URL: <https://zxing.org/w/decode.jsp> (Дата звернення 16.11.2023)
- 25.UnityQRCode [Электронный ресурс] – URL: <https://assetstore.unity.com/packages/tools/integration/qr-code-barcode-scanner-and-generator-cross-platform-pro-56083> (Дата звернення 16.11.2023)
- 26.McConnel S.,Tripp L. Professional Software Engineering: Fact or Fiction? // IEEE SOFTWARE. - 1999. -Nov. - Dec. - P. 13-18
- 27.Fokker J. Functional programming. Dept. of CS. Utrecht University, 1995.
- 28.ARNOLD, Ken; GOSLING, James; HOLMES, David. The Java programming language. Addison Wesley Professional, 2005.
- 29.LUTZ, Mark. Programming python. " O'Reilly Media, Inc.", 2001.
- 30.HEJLSBERG, Anders, et al. The C# programming language. Pearson Education, 2008.
- 31.ELLIOTT, Eric. Programming JavaScript applications: Robust web architecture with node, HTML5, and modern JS libraries. " O'Reilly Media, Inc.", 2014.
- 32.OUALLINE, Steve. Practical C++ programming. " O'Reilly Media, Inc.", 2003.
- 33.NAHAVANDIPOOR, Vandad. IOS 8 Swift Programming Cookbook: Solutions & Examples for IOS Apps. " O'Reilly Media, Inc.", 2014.
- 34.LERDORF, Rasmus; TATROE, Kevin; MACINTYRE, Peter. Programming Php. " O'Reilly Media, Inc.", 2006.
- 35.TILKOV, Stefan; VINOSKI, Steve. Node. js: Using JavaScript to build high-performance network programs. IEEE Internet Computing, 2010, 14.6: 80-83.
- 36.CELKO, Joe. Joe Celko's SQL for smarties: advanced SQL programming. Elsevier, 2010.

- 37.Роботи на складах: 5 прикладів автоматизації [Електронний ресурс] – URL:  
URL: <https://wareteka.com.ua/uk/blog/roboti-na-skladah-prikladi-avtomatizaciyi/> (Дата звернення 10.10.2023)
- 38.Chen, Rongjun, et al. "Adaptive binarization of QR code images for fast automatic sorting in warehouse systems." *Sensors* 19.24 (2019): 5466.
- 39.IBARAKI, Yasuomi. Automation in somatic embryo production. *Progress in Biotechnology*, 2001, 18: 365-374.
- 40.BARGAL, Nilima, et al. PLC based object sorting automation. *International Research Journal of Engineering and Technology (IRJET)*, 2016, 3.07: 103-108.
- 41.ORTIZ-ZACARIAS, Jhon Rodrigo, et al. Simulation of an Automated Sorting System for Peruvian mangoes based on computer vision. In: 2022 5th International Conference on Information and Communications Technology (ICOIACT). IEEE, 2022. p. 385-390.
- 42.FADHIL, A. T.; ABBAR, K. A.; QUSAY, A. M. Computer Vision-Based System for Classification and Sorting Color Objects. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2020. p. 012030.
- 43.Khaing, Moe Win, Aye Mya Win, and Daw Thida Aye. "Automatic Sorting Machine." *international journal of science and engineering applications* 7.8 (2018): 138-142.
- 44.Архітектура програмного забезпечення [Електронний ресурс]. - URL: [http://uk.wikipedia.org/wiki/Архітектура\\_програмного\\_забезпечення](http://uk.wikipedia.org/wiki/Архітектура_програмного_забезпечення). (Дата звернення 17.11.2023)
- 45.Nahavandipoor V. *C# Programming Cookbook / Vandad Nahavandipoor* – O'Reilly, 2016. – 287 с.: ил. – 978-5-8459-3871-2
- 46.Основні відомості про мову програмування C# [Електронний ресурс] – URL: <https://armedsoft.com/ua/blog/osnovni-vidomosti-pro-movu-programuvannya-c> (Дата звернення 11.11.2023)
- 47.Кнастер С#. Програмування / Скотт Кнастер, Вакар Малик, Марк Далрімпл – Діалектика-Вільямс, 2018. – 304 с.: ил. – ISBN 978-5- 8459-1826-0

- 48.OOP Concepts for Beginners: What is Polymorphism [Електронний ресурс] – URL: <https://stackify.com/oop-concept-polymorphism/> (Дата звернення 20.11.2023)
- 49.Єфіменко, Людмила Іванівна, and Михайло Петрович Тиханський. "Моделювання навантаження на опорні конструкції важких стрічкових конвеєрів." (2013).
- 50.Коруняк, П., et al. "Шляхи вдосконалення конструкцій стрічкових конвеєрів." Вісник Львівського національного аграрного університету. Агроінженерні дослідження 18 (2014): 74-78.
- 51.Куроп'ятник, Олексій Сергійович. "Параметрична оптимізація стрічкових конвеєрів за критерієм енергоефективності." (2021).
- 52.Ромасевич, Ю. О. "Аналіз перехідних режимів руху стрічкових конвеєрів." Підйомно-транспортна техніка 1 (2018): 66-75.
- 53.JULIANI, Arthur, et al. Unity: A general platform for intelligent agents. arXiv preprint arXiv:1809.02627, 2018.
- 54.KIM, Sung Lae, et al. Using Unity 3D to facilitate mobile augmented reality game development. In: 2014 IEEE World Forum on Internet of Things (WF-IoT). IEEE, 2014. p. 21-26.
- 55.
- 56.SKOROKHODOV, Dmitriy, et al. Using augmented reality technology to improve the quality of transport services. In: International Conference on Convergent Cognitive Information Technologies. Cham: Springer International Publishing, 2018. p. 339-348.
- 57.The Different Types of Industrial Automation Systems and How They Can Benefit You [Електронний ресурс] – URL: <https://southshorecontrols.com/the-different-types-of-automation-systems-and-how-they-can-benefit-you/> (Дата звернення 25.11.2023)
- 58.Quick Sort [Електронний ресурс] – URL: <https://www.faceprep.in/algorithms/quick-sort/> (Дата звернення 13.11.2023)



- 59.Доровський, Володимир Олексійович. Ідентифікація професійних знань операторів автоматизованих систем управління. Diss. –Херсон. держ. техн. ун-т, 2004.
- 60.Verhagen, Wim JC, et al. "A method for identification of automation potential through modelling of engineering processes and quantification of information waste." *Advanced Engineering Informatics* 29.3 (2015): 307-321.
- 61.Liu, Shih-Chi, and James TP Yao. "Structural identification concept." *Journal of the Structural Division* 104.12 (1978): 1845-1858.
- 62.Wolf, Wayne. "Cyber-physical systems." *Computer* 42.03 (2009): 88-89.
- 63.Lee, Edward A. "The past, present and future of cyber-physical systems: A focus on models." *Sensors* 15.3 (2015): 4837-4869.
- 64.Leitao, Paulo, et al. "Smart agents in industrial cyber–physical systems." *Proceedings of the IEEE* 104.5 (2016): 1086-1101.

Додатки



## 1. Назва та галузь застосування

1.1. Назва – Автоматизована системи сортування об'єктів за QR-кодом.

1.2 Галузь застосування – сортування.

## 2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ №247 від 18.09.2023 р.

## 3. Мета та призначення розробки.

Метою роботи є технологічне оновлення системи сортування об'єктів за QR-кодом на базі рушія Unity.

## 4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Monica Burns. Deeper Learning With QR Codes and Augmented Reality: A Scannable Solution for Your Classroom / Corwin - First edition. – 2016 - с. 5-6, с. 17-18
2. Malburg, L., Seiger, R., Bergmann, R., Weber, B., 2020с. Using Physical Factory Simulation Models for Business Process Management Research, in: Del R'io Ortega, A., Leopold, H., Santoro, F.M. (Eds.), Business Process Management Workshops, Springer. pp. 95–107.
3. Object Detection for Smart Factory Processes by Machine Learning. Procedia Computer Science. / Lukas Malburg, Manfred-Peter, Seiger Ronny, Klein Patrick, Bergmann Ralph. – 2021р.

## 5. Вимоги до розробки.

### 5.1. Перелік головних функцій:

- Генерація QR-коду;
- Сканування та розшифрування QR-коду;
- Сортування об'єктів;

## 5.2. Основні технічні вимоги до розробки.

### 5.2.1. Вимоги до програмної платформи:

- Windows 10/11;
- Середовище для розробки та тестування Unity 3D;
- Мобільна операційна система на базі Android;

### 5.2.2. Умови експлуатації системи:

- робота на мобільних додатках;
- можливість цілодобового функціонування системи;
- дані оновлюються і є актуальними.

## 6. Стадії та етапи розробки.

### 6.1 Пояснювальна записка:

- |  |               |
|--|---------------|
| 1. Аналіз методів сортування                   | 03.10.2023 р. |
| 2. Розробка структури системи                  | 18.10.2023 р. |
| 3. Розробка програмного забезпечення           | 13.11.2023 р. |
| 4. Тестування програмного забезпечення системи | 22.11.2023 р. |

### 6.2 Графічні матеріали:

- |  |               |
|--|---------------|
| 1. Розробка моделі системи             | 20.10.2023 р. |
| 2. Розробка діаграм системи            | 23.10.2023 р. |
| 3. Тестування програмного забезпечення | 25.11.2023 р. |

### 7. Порядок контролю і приймання.

7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до 28 11 2023 р.

7.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до 16 12 2023 р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до 20 12 2023 р.

## Додаток Б – Лістинг програмної частини генератора QR-коду

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using ZXing;
using ZXing.QrCode;
using UnityEngine.UI;
using TMPro;

public class QRgenerator : MonoBehaviour
{
    [SerializeField]
    private RawImage _rawImageReceiver;
    [SerializeField]
    private TMP_InputField _textInputField;
    private Texture2D _storeEncodedTexture;

    // Start is called before the first frame update
    void Start()
    {
        _storeEncodedTexture = new Texture2D(256, 256);
    }

    private Color32[] Encode(string textForEncoding, int width, int height)
    {
        BarcodeWriter writer = new BarcodeWriter
        {
            Format = BarcodeFormat.QR_CODE,
            Options = new QrCodeEncodingOptions
            {
                Height = height,
                Width = width
            }
        };
        return writer.Write(textForEncoding);
    }

    public void OnClickEncode()
    {
        EncodeTextToQrCode();
    }

    private void EncodeTextToQrCode()
    {
        string textWrite = string.IsNullOrEmpty(_textInputField.text)
            ? "You should write something" : _textInputField.text;
        Color32[] _convertPixelTotexture =
            Encode(textWrite, _storeEncodedTexture.width, _storeEncodedTexture.height);
        _storeEncodedTexture.SetPixels32(_convertPixelTotexture);
        _storeEncodedTexture.Apply();
        _rawImageReceiver.texture = _storeEncodedTexture;
    }
}

```

## Додаток В – Лістинг програмної частини сканера QR-коду

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using ZXing;
using TMPro;
using UnityEngine.UI;

public class QrCodeScanner : MonoBehaviour
{
    [SerializeField]
    private RawImage _rawImageBackground;
    [SerializeField]
    private AspectRatioFitter _aspectRatioFitter;
    [SerializeField]
    private TextMeshProUGUI _textOut;
    [SerializeField]
    private RectTransform _scanZone;

    private bool _isCamAvaible;
    private WebCamTexture _cameraTexture;

    // Start is called before the first frame update
    void Start()
    {
        SetUpCamera();
    }

    // Update is called once per frame
    void Update()
    {
        UpdateCameraRender();
    }

    private void SetUpCamera()
    {
        WebCamDevice[] devices = WebCamTexture.devices;

        if (devices.Length == 0)
        {
            _isCamAvaible = false;
            return;
        }
        for (int i = 0; i < devices.Length; i++)
        {
            if (devices[i].isFrontFacing == false)
            {
                _cameraTexture = new WebCamTexture(devices[i].name,
                    (int)_scanZone.rect.width,
                    (int)_scanZone.rect.height);
            }
        }

        _cameraTexture.Play();
        _rawImageBackground.texture = _cameraTexture;
    }
}

```

```
    _isCamAvaible = true;
}

public void UpdateCameraRender()
{
    if (_isCamAvaible == false)
    {
        return;
    }
    float ratio = (float)_cameraTexture.width / (float)_cameraTexture.height;
    _aspectRatioFitter.aspectRatio = ratio;

    int orientation = -_cameraTexture.videoRotationAngle;
    _rawImageBackground.rectTransform.localEulerAngles = new Vector3(0, 0, orientation);
}

public void OnClickScan()
{
    Scan();
}

private void Scan()
{
    try
    {
        IBarcodeReader barcodeReader = new BarcodeReader();
        Result result = barcodeReader.Decode(_cameraTexture.GetPixels32(),
            _cameraTexture.width,
            _cameraTexture.height);
        if (result != null)
        {
            _textOut.text = result.Text;
        }
        else
        {
            _textOut.text = "FAILED TO READ QR CODE";
        }
    }
    catch
    {
        _textOut.text = "FAILED IN TRY";
    }
}
}
```



## Додаток Г – Лістинг програмної частини сортування

```
using System;
using System.IO;

class FileAlphabeticalSort
{
    // Метод для сортування рядків за алфавітом
    static void SortLinesAlphabetically(string filePath, string newFilePath)
    {
        try
        {
            // Читання усього вмісту файлу
            string[] lines = File.ReadAllLines(filePath);

            // Сортування рядків за алфавітом
            Array.Sort(lines, StringComparer.CurrentCulture);

            // Запис відсортованого вмісту у новий файл
            File.WriteAllLines(newFilePath, lines);

            Console.WriteLine("Файл був відсортований та перезаписаний з новою назвою.");
        }
        catch (Exception e)
        {
            Console.WriteLine("Виникла помилка: " + e.Message);
        }
    }

    static void Main()
    {
        string path = "test.txt";
        string newPath = "sortedTest.txt";

        // Виклик методу сортування рядків у файлі за алфавітом з перезаписом у новий файл
        SortLinesAlphabetically(path, newPath);
    }
}
```

**Додаток Г – Ілюстративна частина (обов'язковий)****ІЛЮСТРАТИВНА ЧАСТИНА****Автоматизована система сортування об'єктів за QR-кодом**

1. Мета, предмет та практична цінність дослідження;
2. Головні завдання роботи;
3. Приклад ідентифікації об'єктів на конвеєрних стрічках в промисловості;
4. Схема роботи конвеєра;
5. Діаграма діяльності програмного забезпечення;
6. Алгоритм ідентифікації;
7. Розробка генератора QR-коду;
8. Тестування генератора QR-коду;
9. Розробка сканера QR-коду;
10. Тестування сканера QR-коду;
11. Висновки.

Виконав: студент 2-го курсу, групи  
ЗАКІТ-22м

Андрій БАСАЙ

Керівник: д.т.н., професор каф. КСУ  
Володимир ДУБОВОЙ

« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Опонент: к.т.н., доцент, професор каф.  
АІТ

Руслан БЕЛЗЕЦЬКИЙ

« \_\_\_\_ » \_\_\_\_\_ 2023 р.

# АВТОМАТИЗОВАНА СИСТЕМА СОРТУВАННЯ ОБ'ЄКТІВ ЗА QR-КОДОМ

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

Студент групи ЗАКІТ-22м

Андрій БАСАЙ

Керівник роботи

д.т.н, професор каф. КСУ

Володимир ДУБОВОЙ

## Мета, предмет та практична цінність дослідження

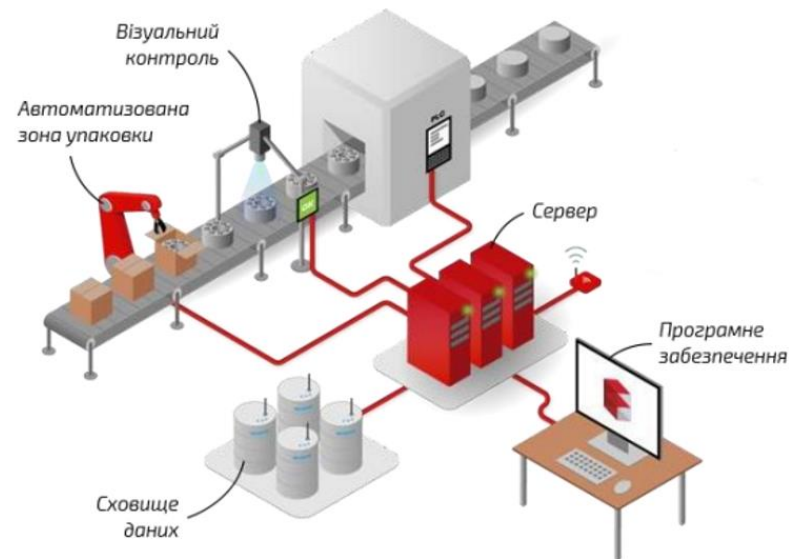
- ▶ **Метою** є технологічне оновлення системи сортування об'єктів за QR-кодом на базі рушія Unity, та оптимізація робочих процесів використовуючи алгоритм швидкого сортування.
- ▶ **Предметом дослідження** є методи та засоби для зчитування та розпізнавання QR-кодів, та подальшої роботи з даними, а саме сортування.
- ▶ **Практична цінність** полягає в удосконаленні програмного забезпечення системи сортування об'єктів за QR-кодами використовуючи сучасний рушій Unity.

## Головні завдання роботи

► **Завдання роботи:**

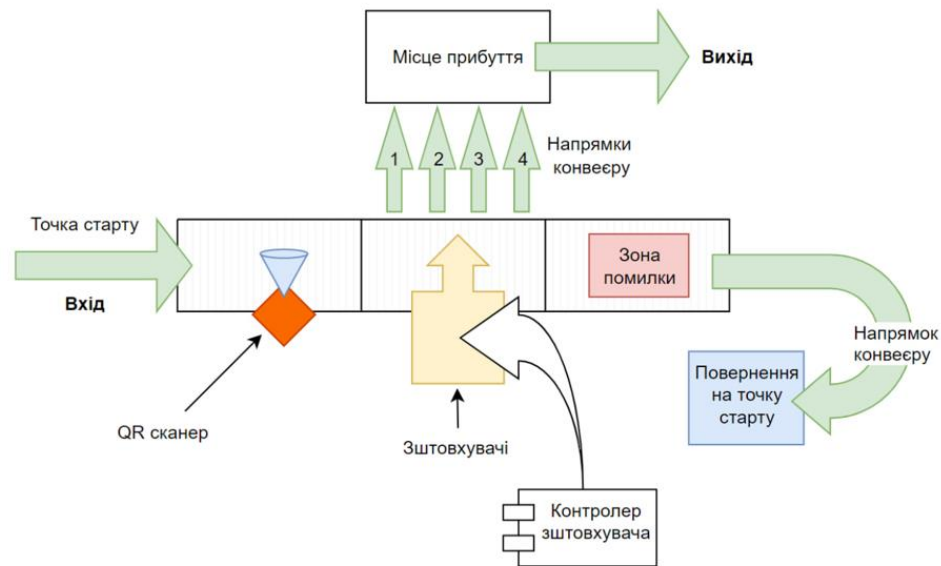
1. аналіз методів сортування об'єктів на конвеєрі;
2. розробка структури системи;
3. розробка програмного забезпечення;
4. тестування та дослідження програмного забезпечення;

## Приклад ідентифікації об'єктів на конвеєрних стрічках в промисловості



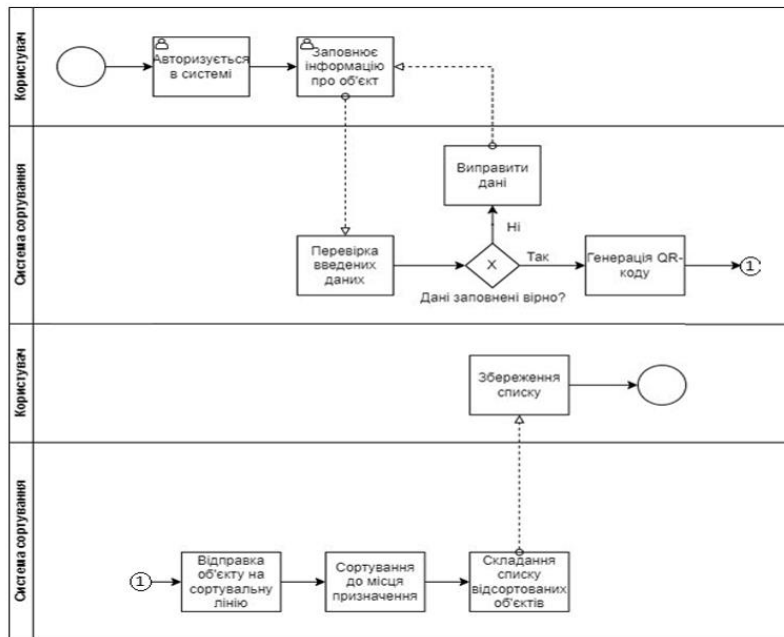
- ▶ Ідентифікація складних і багатограних об'єктів у промисловому виробництві стала головною проблемою при побудові систем управління для цих суб'єктів. Вирішення труднощів ідентифікації є фундаментальною передумовою, яка служить метою для цілеспрямованого проектування систем управління та створення адаптивних структур, здатних працювати в умовах, що характеризуються неповнотою інформації та розмаїттям об'єктів

## Схема роботи конвеєра



- У цілому, конвеєр із QR-сканером є складною системою, яка поєднує в собі технологію сканування та механізми сортування та контролю руху об'єктів з метою ефективного та точного розподілу матеріалів чи товарів.

## Діаграма діяльності програмного забезпечення

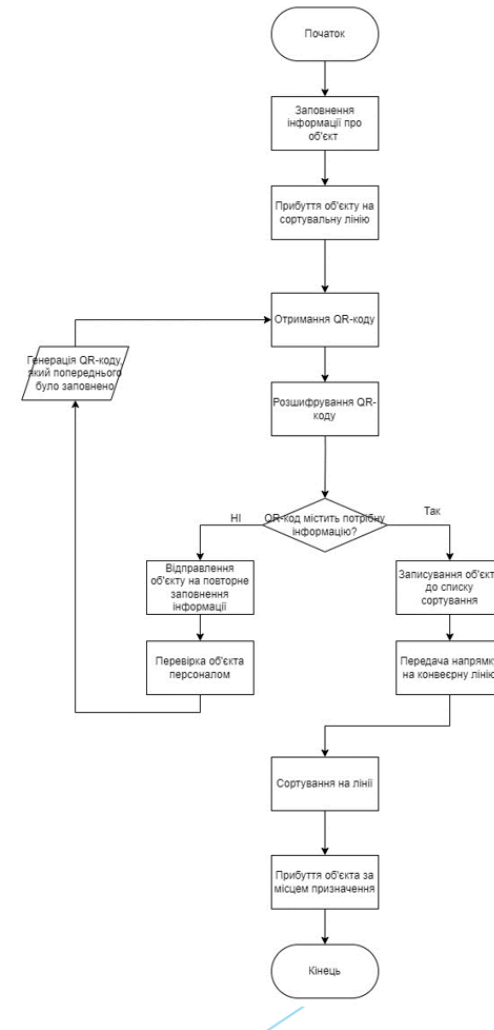


- ▶ Реалізувати логіку сортування на основі інформації, отриманої з QR-кодів.
- ▶ Для зберігання даних про об'єкти та їхнє розташування можна використовувати базу даних або локальне сховище даних.
- ▶ Перед впровадженням системи провести тестування для перевірки правильності зчитування QR-кодів, логіки сортування та коректності роботи інтерфейсу.

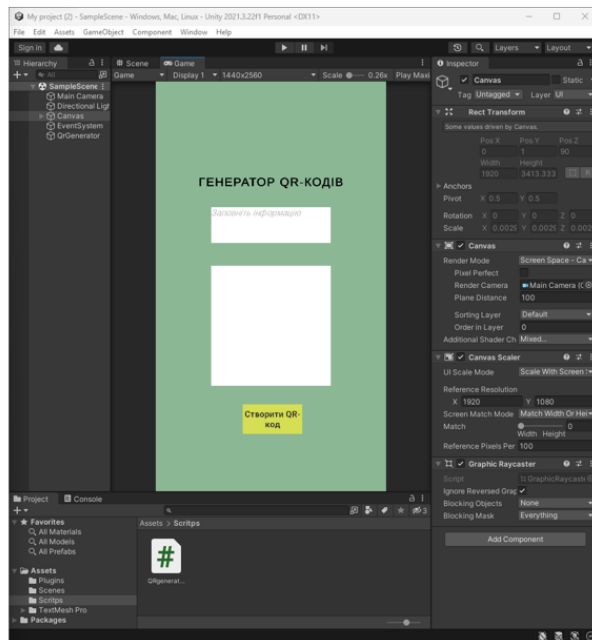


## Алгоритм ідентифікації

- ▶ Особливу увагу в алгоритмічному забезпеченні системи сортування за QR-кодом приділяють виявленню та усуненню можливих помилок у розпізнаванні чи інтерпретації інформації з кодів. Це включає в себе розробку алгоритмів корекції помилок та адаптації до різних умов зчитування QR-кодів.

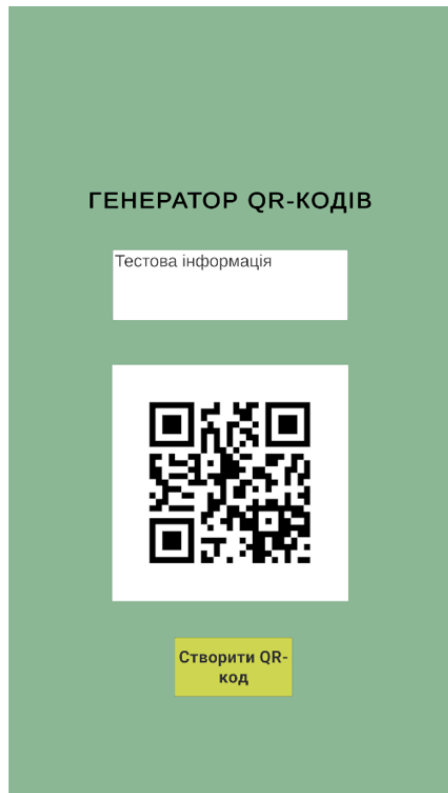


# Розробка генератора QR-кодів



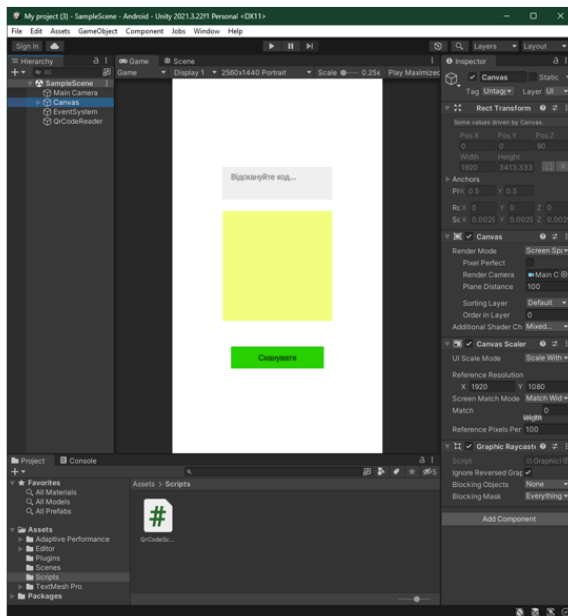
- ▶ При розробці інтерфейсу для генерації QR-коду у середовищі Unity, враховувано основні принципи дизайну користувацького інтерфейсу для забезпечення зручності та ефективності користувачів.
- ▶ Цей інтерфейс містить елементи керування, що спрощують процес створення QR-коду, включаючи поля для введення даних, які закодовані, та кнопку для запуску процесу генерації коду.

## Тестування генератора QR-кодів



- ▶ Системні тести виконуються для перевірки програми в цілому. Це випробування всієї функціональності програми в реальних або симульованих умовах використання, щоб переконатися, що вона працює вірно та задовольняє всі вимоги та очікування користувача.
- ▶ Загалом, ефективна перевірка програми забезпечує надійність, функціональність та задоволення потреб користувача, що робить її ключовим етапом в процесі розробки програмного забезпечення.

# Розробка сканера QR-коду



- ▶ Розробка сканера QR-кодів у середовищі Unity полягає у створенні елементів користувацького інтерфейсу, що забезпечують зручну та ефективну взаємодію користувача з функціоналом сканування QR-кодів. Це включає в себе створення візуальних компонентів, які дозволяють активувати камеру, показувати зображення з неї та обробляти інформацію, отриману з QR-кодів.

## Тестування сканера QR-коду



- ▶ Дослідження також враховує вплив різних параметрів, таких як роздільна здатність камери, на результати сканування.

## ВИСНОВКИ

- ▶ В результаті виконання магістерської роботи було проаналізовано методи сортування об'єктів на конвеєрі, розроблено структуру системи, її програмне забезпечення. Було проведено тестування та дослідження програмного забезпечення.
- ▶ Під час виконання роботи було розроблено та досліджено програмне забезпечення, а саме - алгоритми сортування, програма для генерації QR-кодів та сканер для читання QR-кодів. Було виявлено, що головними перевагами є швидкість зчитування, в своїй структурі QR-коди мають вбудованій структурі і коректуючі коди, можливість містити в одному коді багато інформації.
- ▶ Результати дослідження можуть бути використані для подальшого розвитку технологій сортування, що використовують QR-коди. Шляхом вдосконалення алгоритмів обробки даних та розширення можливостей інтеграції з іншими технологіями, можливо досягти ще вищого рівня ефективності та автоматизації в різних галузях.
- ▶ Загалом, використання автоматизованих систем сортування за QR-кодами є перспективним та дієвим рішенням для підприємств, що мають зацікавленість у підвищенні ефективності сортувальних та виробничих процесів, оптимізації обслуговування та зниженні витрат.

Додаток Д – Протокол перевірки кваліфікаційної роботи (обов'язковий)

**ПРОТОКОЛ  
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: «Автоматизована система сортування об'єктів за QR-кодом»

Тип роботи: Магістерська кваліфікаційна робота  
(БДР, МКР)

Підрозділ АПТ, ФІТА  
(кафедра, факультет)

**Показники звіту подібності Unicheck**

Оригінальність 98.9% Схожість 1.1%

Аналіз звіту подібності (відмітити потрібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку Роман МАСЛІЙ  
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи Андрій БАСАЙ  
(підпис) (прізвище, ініціали)

Керівник роботи Володимир ДУБОВОЙ  
(підпис) (прізвище, ініціали)