

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційних радіотехнологій та технічного захисту інформації
(повна назва)

Кафедра медіаінженерії та інформаційних радіоелектронних систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
(позначення документа)

Методи створення ігор в режимі онлайн
з можливістю взаємодії із сервером

(тема)

Виконав:

студент 2 курсу, групи МІМ-22-1
Богдан ПОНОМАРЕНКО

(прізвище, ініціали)

Спеціальність 172 Телекомунікації та радіотехніка

(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Медіаінженерія
(повна назва освітньої програми)

Керівник доц. Роман ЦЕХМІСТРО
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис) Володимир КАРТАШОВ
(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій та технічного захисту інформації
Кафедра Медіаінженерії та інформаційних радіоелектронних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 172 Телекомунікації та радіотехніка
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма "Медіаінженерія"

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 20 ____ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Пономаренку Богдану Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи створення ігор в режимі онлайн з можливістю взаємодії із сервером

затверджена наказом по університету від " 20 " 10 2023 р. № 1224 СТ

2. Термін подання студентом роботи 08.01.2024 р.

3. Вихідні дані до проекту (роботи)

1. Створення концепт-документу гри,

2. Розробка алгоритму та гри з власною базою даних;

3. Організація та взаємодії з сервером.

4. Перелік питань, що потрібно опрацювати в роботі

ВСТУП

1. Аналіз та дослідження засобів створення ігрового проекту

2. Розробка концепт-документу гри Snake CLASSIC

3. Розробка ігрового рівня гри Snake CLASSIC

4. Розробка та організація взаємодії її з сервером online гри

ВИСНОВКИ

ПЕРЕЛІК ПОСИЛАНЬ

ДОДАТКИ

5. Перелік графічного матеріалу із зазначенням обов'язкових креслеників, схем, плакатів, комп'ютерних ілюстрацій:

1) Актуальність роботи, 2) Аналіз та дослідження засобів створення ігрового проекту, 3)

Аналіз та обґрунтування вибору платформи та мови програмування, 4) Розробка концепт-

документу гри, розробка сюжету, 5) Розробка та опис ігрової графіки, 6) Аналіз жанру гри, 7)

Розробка алгоритму гри SNAKE CLASSIC, 8) Розробка бази даних, 9) Оптимізація та

тестування, 10) Етапи тестування, 11) Висновки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз та дослідження засобів створення ігрового проекту	20.10.23–28.11.23	
2.	Розробка концепт-документу гри Snake	21.11.23–28.11.23	
3.	Розробка ігрового рівня гри Snake	23.11.23–02.12.23	
4.	Розробка та організація взаємодії її з сервером online гри	01.12.23–15.12.23	
5.	Графічна частина роботи	15.12.23–20.12.23	
6.	Перевірка керівником	20.12.23-24.12.23	
7.	Перевірка на академічний плагіат	24.12.23-26.12.23	
8.	Перевірка завідувачем кафедри, рецензування	27.12.23-31.12.23	

Дата видачі завдання _____ 20.10.2023 р.

Студент _____ Пономаренко БОГДАН
(підпис)

Керівник роботи _____ доц. Роман ЦЕХМІСТРО
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи має: 66 с., 33 рис., 1 табл., 58 джерел.

JS, PHP, CSS3, HTML5, CANVAS, UNITY, MYSQL, SQL, VISUAL STUDIO CODE (VSC), СЕРВЕР, КЛІЄНТ, РЕГІСТРАЦІЙНА ФОРМА, ПОЛЕ ВІДНОВЛЕННЯ ПАРОЛЮ, ВХІД ТА ВИХІД ІЗ СИСТЕМИ.

Об'єкт дослідження – online ігри з взаємодією з сервером.

Предмет дослідження – інструменти створення online ігор.

Мета кваліфікаційної роботи – метою кваліфікаційної роботи є вивчення можливостей створення online ігор, дослідження підходів створення гри.

Методи дослідження – теоретичний аналіз, числові розрахунки, математичне моделювання, статистична обробка даних.

У процесі роботи розглянуті кроки створення проекту online ігрового контенту з можливістю використання взаємодії із сервером. У даній роботі наведено детальний аналіз кожного з методів створення online ігор, опис функціонування та процес створення однієї з ігор.

ABSTRACT

The explanatory note of the qualification work has: 66 pages, 33 figures, 1 table, 58 sources.

JS, PHP, CSS3, HTML5, CANVAS, UNITY, MYSQL, SQL, VISUAL STUDIO CODE (VSC), SERVER, CLIENT, REGISTRATION FORM, PASSWORD RECOVERY FIELD, LOGIN AND LOGIN.

The object of research is online games, interaction with the server.

The subject of research is tools for creating online games in programming languages.

The purpose of the qualification work – The purpose of this diploma is to study the possibilities of creating online games, and the right approach to create a game, from your own survey to the budget, choosing the technology on which the game will be created.

Research methods - Studying the main stages of creating games using programming languages, creating your own game with a database, registration field, password recovery, login, and logout.

In the course of work, the steps of creating a project on how to properly develop an online game with the possibility of using interaction with the server are considered. This work provides a detailed analysis of each of the methods of creating online games, a description of the functioning and the process of creating one of the games.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

Apache - це відкритий веб-сервер-програма, який грає ключову роль у розгортанні веб-сайтів та веб-додатків;

Canvas - це елемент HTML5, який надає можливість малювати графіку на веб-сторінці за допомогою скриптів JavaScript;

CSS (Cascading Style Sheets) - це мова опису стилів, яка використовується для задання вигляду і форматування веб-документів, «каскадні таблиці стилів»;

GitHub - це веб-платформа для зберігання та спільної роботи над проектами з використанням системи керування версіями Git;

HTML (Hypertext Markup Language) - це стандартна мова розмітки для створення та відображення веб-сторінок;

JavaScript - це високорівнева мова програмування, яка широко використовується для створення інтерактивних веб-сторінок;

PHP (Hypertext Preprocessor) - це скриптова мова програмування, яка використовується для розробки веб-застосунків;

SQL (Structured Query Language) — це мова запитів для роботи з реляційними базами даних;

Unity - графічний двигун для створення ігор та інших інтерактивних віртуальних середовищ;

VSC - (Visual Studio Code) редактор вихідного коду;

БД - База даних;

СУБД MySQL (MySQL Database) - це система управління базами даних, яка використовується для зберігання, організації та управління даними.

ЗМІСТ

Перелік умовних скорочень	6
Вступ	9
1 Аналіз та дослідження засобів створення ігрового проекту.....	10
1.1 Аналіз програмно-апаратних засобів для створення ігор з використанням серверних технологій.....	10
1.2 Аналіз та обґрунтування вибору платформи та мови програмування ..	15
1.3 Аналіз та обґрунтування вибору ігрових рушіїв.....	17
1.3.1 Аналіз ігрового двигуна Unity	19
1.3.2 Аналіз ігрового двигуна Unreal Engine	21
1.3.3 Аналіз ігрового двигуна Source 2	23
1.4 Висновки до розділу	24
2 Розробка концепт-документу гри Snake	25
2.1 Розробка сюжету.....	25
2.3 Мета та цільова аудиторія.....	26
2.4 Розробка та опис ігроладу.....	27
2.5 Аналіз жанру гри.....	28
2.6 Висновки до розділу	28
3 Розробка ігрового рівня гри Snake.....	29
3.1 Розробка узагальненого алгоритму ігрового рівня	29
3.2 Розробка локацій ігрового контенту	31
3.3 Етапи розробки проекту.....	33
3.4 Висновки до розділу	39
4 Розробка та організація взаємодії її з сервером online гри	41
4.1 Розробка серверної частини.....	42

4.2 Розробка бази даних.....	46
4.3 Аутентифікація та безпека	51
4.4 Оптимізація та тестування	53
4.5 Висновки до розділу	55
Висновки	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	57
Додатки.....	67
Додаток А.....	68
Додаток Б.....	83
Скріни роботи програми.....	83
Додаток В.....	85
Додаток Г.....	95

ВСТУП

Метою кваліфікаційної роботи є розгляд можливостей створення online ігор зі взаємодією з сервером.

Зараз, онлайн ігри досягли вражаючих висот, існує велика кількість компаній, наприклад Valve, CD PROJEKT RED, 4A Games, GSC Game World, які займаються розробками ігор, є кіберспортивні змагання, де розігруються мільйони доларів, більшість людей йдуть в ігри. Багато гравців воліють грати в мережеві ігри (online), наприклад в багатокористувацькі онлайн-ігри (ММО), такі як дота 2, шутери, бойовики, спортивні симулятори або інші жанри.

На сьогоднішній день онлайн-ігри є дуже актуальними і популярними в ігровій індустрії. Наприклад, онлайн-ігри дозволяють гравцям спілкуватися, взаємодіяти та конкурувати з іншими гравцями з усього світу. Це надає соціальний аспект, який багато гравців високо цінують. Також, в онлайн іграх пропонують онлайн-режим, де гравці можуть спільно виконувати завдання, боротися проти інших гравців або створювати великі онлайн-спільноти. Із розвитком технологій і розповсюдженням інтернету, доступ до онлайн-ігор став широко доступним. Гравці можуть грати на різних платформах, включаючи ПК, консолі та мобільні пристрої.

Проте актуальність онлайн-ігор може змінюватися залежно від ринків і вибору споживачів, такі як розвиток технологічних інновацій (наприклад, віртуальної реальності та т.і.), нові тенденції в геймдизайні та суспільні події, також можуть впливати на популярність певних типів онлайн-ігор.

1 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ЗАСОБІВ СТВОРЕННЯ ІГРОВОГО ПРОЕКТУ

1.1 Аналіз програмно-апаратних засобів для створення ігор з використанням серверних технологій

Методи створення ігор з використанням серверних технологій - це різноманітні підходи та техніки, які розробники використовують для створення високоякісних ігрових контентів з використанням серверів.

Методи створення ігор з серверними технологіями:

- мережева архітектура , клієнт-сервер ,одна з найпоширеніших архітектур, де клієнти взаємодіють із сервером для отримання актуальної інформації та надсилання даних. Це дозволяє централізоване управління грою;
- масові багатокористувацькі онлайн ігри (ММОГ) – це ігрові світи, тобто , розробка великих віртуальних світів, де тисячі гравців можуть взаємодіяти одночасно. Тут використовують сервери для забезпечення спільної гри та обміну даними;
- ігрові сервери - логіка гри на сервері, розміщення важливої логіки гри на сервері для уникнення підвищення можливостей обману та гарантування централізованого контролю над ігровим процесом;
- хмарні технології - використання хмарних технологій для забезпечення послуг, таких як зберігання даних гри, мережева інтеграція та обробка даних;
- безпека та аутентифікація - впровадження заходів безпеки для захисту від шахрайства та забезпечення конфіденційності даних гравців.

Ці методи можуть бути використані окремо чи в комбінації, залежно від конкретних вимог і характеристик ігрового проекту. В моїй роботі використовується метод , мережева архітектура та безпеку та аутентифікацію.

Тому що, в роботі потрібен запис даних до сервера, створення аккаунта , запобігання від викрадення даних так її повна конфіденційність.

Але, для створення власної гри цього методу цього мало, потрібно ще визначитися з іншими кроками, які описані внизу. Розглянемо кроки та технології, які можуть бути використані.

Кроки для створення власної гри:

1. Вибір платформи та мови програмування.

Ігри бувають для різних платформ, наприклад, online, мобільні пристрої, ПК. В ході роботи були розглянуті літературні джерела, де автори рекомендують в основному користуватись такими мовами програмування як: C++, JavaScript, C#, Python, Java та інші. В проекті використовується JS, ось його головна характеристика.

Сучасний JavaScript визначається як мова програмування, яка пропонує високий рівень безпеки. Вона не надає низькорівневий доступ до пам'яті або процесора, оскільки спочатку розроблялася для використання у веб-браузерах, які не вимагають такого рівня доступу.

Можливості JS залежать від оточення, в якому він працює. Наприклад, Node.JS підтримує функції читання/запису довільних файлів, виконання мережевих запитів тощо.

У браузері для JavaScript доступне все, що пов'язане з маніпулюванням веб-сторінками, взаємодією з користувачем та веб-сервером.

Наприклад, у браузері JS може:

- додавати новий HTML-код на сторінку, змінювати наявний вміст, модифікувати стилі;
- реагувати на дії користувача, натискання миші, переміщення вказівника, натискання клавіш;
- запам'ятовувати дані на стороні клієнта ("local storage");
- отримувати та встановлювати куки, ставити запитання відвідувачу, показувати повідомлення;

- надсилати мережеві запити на віддалені сервери, завантажувати та завантажувати файли (технології AJAX та COMET).

Особливість JS в проекті:

1. Повна інтеграція з HTML та CSS;
2. Простий синтаксис;
3. Він може оживити сторінку;
4. Легка взаємодія з серверною мовою PHP;
5. Підтримка скриптів для всіх популярних браузерів;
6. Швидкість роботи і продуктивність.

2. Розробка серверної частини.

Розробка сервера для гри, це одна з найголовніших речей в створенні, потрібно правильно записувати данні та перевіряти в базі даних. Існують такі технології, як Node.js, PHP, Python (з використанням бібліотек Flask або Django), ASP.NET, Java (з використанням Spring або других фреймворків). Для цієї роботи, обрано мову програмування PHP, тому що:

1. Ідеально підходить для передачі даних.
2. Прямо може виконувати запити.
3. PHP легкий у навчанні та зрозумілий синтаксис.
4. Сумісний з базою даних.

За допомогою PHP створено форму входу та виходу, поле реєстрації та відновлення пароля. Він також обробляє дані та перевіряє їх. PHP дуже популярна серверна мова програмування, на ній створені такі сайти як, Facebook, Wikipedia, Youtube, пошукова система YANOO.

3. Робота з базою даних.

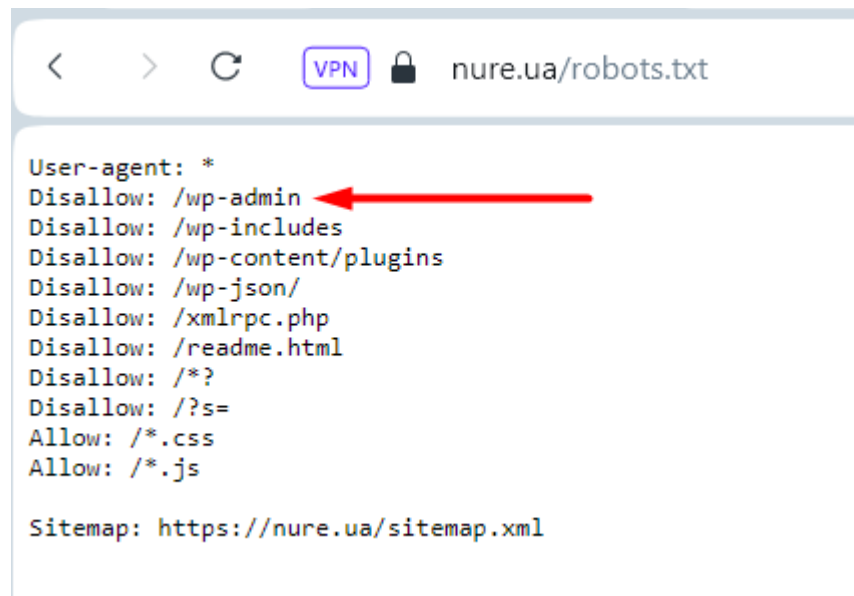
Для запису для збереження даних ігрока, ігрового процесу та іншої інформації використовують бази даних на мові SQL, MySQL, PostgreSQL, MongoDB, Oracle - це найпопулярніші бази даних.

В моїй роботі було використано базу даних MySQL, вона – одна з найпоширеніших систем керування базами даних.

MySQL - це безкоштовна система управління реляційними базами даних. Розробку та підтримку MySQL здійснює корпорація Oracle, яка отримала права на торгову марку разом із поглиненою Sun Microsystems, яка раніше придбала шведську компанію MySQL AB.

Розглянемо чому саме MySQL було обрано для створення проєкту кваліфікаційної роботи:

- MySQL є популярним вибором для зберігання даних веб-застосунків. Багато веб-сайтів та онлайн-сервісів, таких як блоги, інтернет-магазини та соціальні мережі, використовують MySQL для управління своїми базами даних;
- її використовують для збереження інформації, для публікацій статей, дані користувачів, налаштуваннях сайтів та інших інформацій. Наприклад, сайт нашого університету написаний на WordPress(WP) (рис. 1.2);



```

User-agent: *
Disallow: /wp-admin
Disallow: /wp-includes
Disallow: /wp-content/plugins
Disallow: /wp-json/
Disallow: /xmlrpc.php
Disallow: /readme.html
Disallow: /*?
Disallow: /*s=
Allow: /*.css
Allow: /*.js

Sitemap: https://nure.ua/sitemap.xml

```

Рисунок 1.2 - Перевірка сайту nure.ua на наявність CMS

- вона застосовується для зберігання даних про гравців, їх досягнень, рейтингів та інші ігрові параметрів;
- багато навчальних сайтів, використовують MySQL, для зберігання даних учнів, внесення їх оцінок, відвідуваності та успішності, здачі та

зберігання домашніх завдань, практичних робіт також лабораторних та курсових;

- системи управління персоналом та CRM-системи, можуть використовувати MySQL для зберігання та обробки даних;
- відкритий код, в MySQL є відкритим програмним забезпеченням, що означає, що ви можете вільно користуватися, модифікувати та розповсюджувати код;
- онлайн-магазини та платформи електронної комерції часто використовують MySQL для управління продуктами, замовленнями, клієнтськими даними та іншою пов'язаною інформацією.

4. Аутентифікація та безпека.

Реалізація механізмів аутентифікації та авторизації для забезпечення безпеки гри та захисту від вторгнень та викрадень персональних даних. Цю функцію також виконує MySQL. MySQL, як система управління базами даних (СУБД), надає кілька рівнів захисту для забезпечення безпеки даних та запобігання несанкціонованому доступу.

Механізми безпеки в MySQL:

- шифрування даних - MySQL надає можливість зашифрувати дані під час їх передачі між клієнтом та сервером, використовуючи протокол SSL/TLS;
- аутентифікація та авторизація - MySQL використовує систему автентифікації для автентифікації користувачів, засновану на паролях або інших методах тобто, програміст або інший користувач, не може редагувати, що завгодно, для цього на всіх базах даних є паролі;
- SQL-ін'єкції - MySQL забезпечує засоби захисту від SQL-ін'єкцій, такі як використання параметризованих запитів та підготовлених виразів;
- протоколювання та моніторинг – MySQL надає журнал подій, в яких зазначено час проведення робіт, спроби входу, зміни паролю та виконання інших запитів.

1.2 Аналіз та обґрунтування вибору платформи та мови програмування

Вибір платформи на який буде створюватись гра дуже складний процес та залежить від деяких факторів, таких як:

1. Досвід та навички розробників гри.
2. Вид гри.
3. Бюджет.
4. Мультиплатформенність.
5. Ігрові рушії.
6. Цільова аудиторія.

Вид гри, тобто, 2D гра або 3D гра. Для простих 2D ігор може бути вигідно використовувати мови та платформи, що спеціалізуються на цьому, наприклад, Unity з C# для 2D. Для складніших 3D проєктів може підійти Unreal Engine з C++. На будь якій мові можна написати гру, але, все залежить, від типу задачі, яка буде покладена на програміста.

Переваги найпопулярніших мов програмування:

- C++ та C - ці мови вважаються одними з найшвидших, оскільки компілятори C++ та C генерують ефективний машинний код;
- Java - відмінно працює завдяки великій кількості оптимізації віртуальної машини Java;
- C# - використовується в ігровій індустрії та деяких веб-застосунках, таких як ASP.NET. Має добру продуктивність;
- Python - хоча Python є менш продуктивним порівняно з C/C++ чи Java, він часто використовується завдяки зручності та легкості використання;
- JavaScript - використовується на боковому (клієнтському) рівні для веб-розробки. Поширений та швидко виконується завдяки вдосконаленням у двигунах JS, таким як V8;

- PHP - використовується для веб-розробки. Завдяки веб-серверам та кешуванню може бути ефективним для обробки веб-запитів.

Розглянемо найпопулярніші з ігор (рис. 1.3).

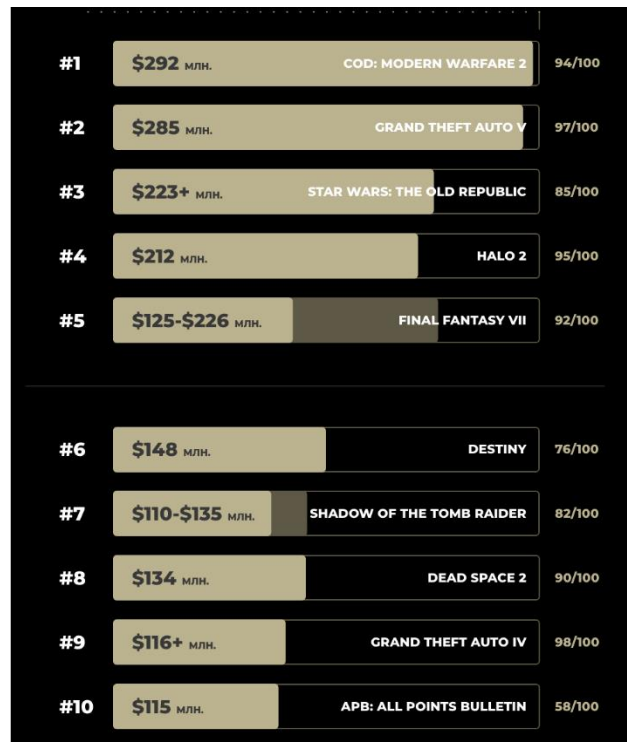


Рисунок 1.3 Топ 10 найдорожчих ігор за увесь час

Перше місце, Call of Duty: Modern Warfare 2, комп'ютерна гра у жанрі шутера від першої особи, розроблена компанією Infinity Ward. Гра випущена 28 жовтня 2022 для персональних комп'ютерів, PlayStation 4, Xbox One, PlayStation 5 і Xbox Series X/S на ігровому рушії IW Engine. По першій грі відразу можна зрозуміти, що людям подобається шутери, стрілялки від першої особи, в неї вклали для створення 292млн \$. І це, не тільки зарплати робітникам, програмістам, це також, рекламні контракти, в цілому реклама, створення графіки, опрацювання всіх поставлених цілей, окрім створення одиночної гри та підтримка, також, многокористувацького режиму, підтримки серверів. В цілому гра отримала переважно хвалебні відгуки.

Мультиплатформенність – гру можуть одразу створювати для всіх видів пристроїв. Для ПК та для телефонів. При цьому гравці можуть

використовувати різні операційні системи, наприклад, Android, iOS або Windows. Найпопулярніші кросплатформені ігри це Minecraft, Genshin Impact.

Ігрові рушії – мають базове програмне забезпечення для комп'ютерної чи відеогри. Під цим поняттям розглядається програмне забезпечення, придатне для повторного використання та розширення, та таким чином може бути розглянуте як основа для розробки різноманітних ігор без суттєвих змін.

Визначення цільової аудиторії для гри є важливим етапом у розробці та маркетингу. Тому, для початку, виконують такі кроки, для визначення цільової аудиторії:

1. Аналіз жанру та стиль гри.
2. Визначення вікової та демографічної групи.
3. Платформи та технічні специфікації.
4. Особливість власної гри.
5. Проведення тестових опитувань, для покращення товариства.

1.3 Аналіз та обґрунтування вибору ігрових рушіїв

Ігровий двигун (game engine) – це базове програмне забезпечення, на основі якого будуються всі інші складові гри. Програмний код, який може використовуватися для створення варіацій гри, аддонов до неї або навіть зовсім нового ігрового світу.

За допомогою ігрових двигунів, можна покращити та об'єднати процес написання гри та створення ігрового процесу. В сучасних проектах використовують ігрові рушії, такі як, Unity, Source, Source 2, Unreal Engine, Godot та інші. Детальний опис кожного з ігрових рушіїв.

В цілому в кожному ігровому движку свої основні вимоги та переваги, в цілому, ось виділення основних переваг та недоліків всіх ігрових рушіїв.

Недоліки та переваги ігрових рушіїв.

Недоліки:

- обмеження у гнучкості та контролю, тобто, ігрові рушії надають безліч інструментів, але можуть бути занадто обмежені в тому, як можна реалізувати певні функції чи ефекти;
- продуктивність. Деякі ігрові двигуни можуть мати проблеми з продуктивністю, особливо при створенні великих і складних ігор;
- розмір файлу та вимоги до ресурсів. Проекти можуть багато важити, через вбудовані бібліотеки, що збільшують обсяг гри;
- залежність від оновлень. Сучасні ігрові рушії регулярно оновлюються, може бути така проблема, як, проблема сумісності;
- ліцензійні обмеження – Постійна платна підписка є в деяких ігрових двигунах;
- складність навчання. Є двигуни які дуже складні для освоєння, особливо для новачків.

Також багато популярних компаній створюють свої ігрові движки, для збереження конфіденційності компанії та незалежного програмного забезпечення.

Переваги:

- прискорення розробки. Ігрові рушії надають готові інструменти та бібліотеки, що значно прискорює процес розробки ігор.;
- графіка та анімація – в багатьох ігрових двигунах, є інструменти для роботи з графікою та анімацією. за допомогою їх можна створювати зручні засоби створення тривимірних моделей, текстур, освітлення та ефектів, що дозволяє створювати візуально вражаючі ігри.;
- крос-платформність – багато ігрових движків підтримують розробку для різних платформ, таких як ПК, консолі, мобільні пристрої та веб.;
- ресурси та інструменти. Багато варіантів рішень того чи іншого поставленого завдання;
- економія ресурсів. Використання ігрових движків може бути економічно ефективнішим, особливо для невеликих студій або

незалежних розробників, оскільки це дозволяє уникнути створення всієї інфраструктури з нуля.

Godot – це відкритий кросплатформер для 2D та 3D ігрових двигунів. Середина розробки запускається на Linux, Android, Windows, HTML5, macOS, BSD та Haiku і може експортувати ігрові проекти на ПК, консолі, мобільні та веб-платформи.

Frostbite - це високопродуктивний і потужний ігровий рушій, про який знають всі фанати ігор від Electronic Arts. Він використовується в багатьох іграх цієї студії, включаючи Battlefield, FIFA, DragonAge, PayBack, Need for Speed та інші. Простіше кажучи, це платформа-універсал, на основі якої написані дій, РПГ, гонки, спортивні симулятори і т.д.

Основні характеристики і особливості рушія Frostbite включають:

- графіка високої якості;
- реалістична фізика;
- професійний та реалістичний звук;
- зручність розробки;
- мультиплатформленість;
- підтримка великих світів;
- онлайн-ігри.

Кожний ігровий движок вмістить в себе ігрову логіку, фізику об'єктів, Геймплей, правила відтворення об'єктів, тощо. Всі інші компоненти гри вписуються "поверх" ігрового движка, і цих компонентів багато. Таким чином, навіть за умови використання одного і того ж ігрового движка, віртуальні світи можуть значно відрізнятися.

1.3.1 Аналіз ігрового движка Unity

Unity – це кросплатформове середовище розробки комп'ютерних ігор. Дозволяє створювати програми, на будь яких платформах, таких як, ПК,

консольні платформи, мобільні телефони, в веб-браузері. Основними перевагами Unity є наявність візуального середовища розробки, міжплатформної підтримки та модульної системи компонентів. До недоліків відносять появу складнощів при роботі з багатокомпонентними схемами та утруднення при підключенні зовнішніх бібліотек.

Переваги Unity:

- мультиплатформеність, дозволяє розробку ігор для різних платформ, наприклад, MacOS, iOS, Windows, Android та інші;
- взаємодія з кількома мовами програмування, є можливість взаємодії наприклад C# та JavaScript;
- є візуальний редактор, який дозволяє розробникам легко створювати та редагувати сцени, анімації, матеріали та інші елементи гри без написання коду.;
- інтеграція 3d та 2d графіки, підтримує розробку як тривимірних, так і двовимірних ігор;
- фізика та анімація, має вбудовані системи фізики та анімацій;
- розширюваність, дозволяє розробникам використовувати сторонні розширення та активи, що полегшує використання сторонніх ресурсів та розширення можливостей ігрового движка.

Недоліки:

- великий розмір бібліотек, Unity має велику бібліотеку, яка може призвести до зайвого обсягу встановленого програмного забезпечення для проектів, які не використовують усі її функції. Це може вплинути на обсяги пам'яті та час завантаження гри;
- проблеми з оптимізацією, для деяких проектів, особливо для дуже великих чи ресурсоємних ігор, може бути складно досягти оптимальної продуктивності;
- велика кількість оновлень, Unity регулярно випускає нові версії та оновлення, що може призводити до того, що розробники повинні витрачати час на адаптацію своїх проектів до нових версій;

- проблеми зі сумісністю , у деяких випадках може виникнути проблеми зі сумісністю між різними версіями Unity, особливо коли використовуються сторонні розширення та активи.

Розглянемо технічні вимоги для встановлення Unity на комп'ютері:

- система оперативної пам'яті (RAM)- мінімум 8 ГБ RAM, рекомендовано 16 ГБ або більше RAM;
- графічна карта (GPU) мінімум NVIDIA GeForce GTX 470 або AMD Radeon 6870 HD з підтримкою DirectX 11, рекомендовано графічна NVIDIA GeForce GTX 970 або AMD Radeon R9 290 або вище;
- DirectX, версія DirectX 11 або новіша;
- місце на жорсткому диску , мінімум 100ГБ;
- процесор (CPU) мінімум двоядерний процесор з частотою 2,5 ГГц, рекомендовано чотирьохядерний процесор з частотою 3,5 ГГц або вище;
- операційна система Windows 10 та вище, macOS 10.14.6+ та вище, Linux Ubuntu 18.04 (LTS) або новіша.

1.3.2 Аналіз ігрового двигуна Unreal Engine

Unreal Engine 5 - ігровий рушій, що розробляється і підтримується компанією Epic Games. Unreal Engine, незалежно від його версії – сама крута і найпопулярніша платформа для початківців розробників.

Unreal Engine 5, як інші версії цього геймдев-двигуна, має кілька переваг:

- графіка та реалізм, UE використовує технології які дозволяють відобразити велику кількість деталей моделі без значного впливу на продуктивність;
- великий функціонал, в Unreal Engine є графічна система програмування, яка полегшує створення ігрової логіки та взаємодії без необхідності в програмуванні;

- кросплатформленість, дозволяє розробляти ігри на всі види платформ, включаючи, РС, мобільні пристрої, консолі і т.і.;
- реалістичний звук, система MetaSounds дозволяє створювати багатошаровий та реалістичний звук, який впливає на гравця і змушує зануритися в гру.

Особливість цієї серії розробки, що ви можете використовувати її безкоштовно, до тих пір, поки, ваш чистий дохід з вашої гри, не буде перебільшувати 3000\$.

Переваги UE:

- графіка та рендеринг , Unreal Engine славиться своєю потужною системою рендерингу, що дозволяє створювати вражаючі візуальні ефекти та реалістичну графіку;
- мультиплатформленість , Забезпечує можливість розробки ігор для різних платформ, таких як ПК, консолі, мобільні пристрої та віртуальна реальність;
- інструменти для розробки, має інтегровані інструменти для створення гри, редагування сцен, анімації, роботи зі звуком та багато інших;
- розширюваність , дозволяє розробникам використовувати сторонні розширення та активи, що покращує та розширює функціональність;
- фізика та анімація , включає в себе розвинену систему фізики, анімації та динамічних властивостей, що дозволяє створювати реалістичні рухи та поведінку об'єктів.

Недоліки UE:

- мова програмування, використовує тільки мову програмування C++ або Blueprint;
- великий обсяг файлів проекту, проекти у Unreal Engine можуть мати великий обсяг файлів, що може ускладнити обробку та роботу з ними;
- важко оптимізувати для мобільних пристроїв , розробка графічно високоякісних ігор для мобільних пристроїв у Unreal Engine може бути складною та вимагати великого обсягу ресурсів.

1.3.3 Аналіз ігрового двигуна Source 2

Source 2 - це ігровий движок, розроблений компанією Valve Corporation. Цей движок використовується для створення ігор та віртуальної реальності.

Переваги Source 2:

- висока якість графіки, Source 2 вражає високоякісною графікою та реалістичними візуальними ефектами. Він підтримує сучасні графічні технології та оснащений потужним рендерингом;
- широкий функціонал для розробників, Source 2 надає розробникам великий набір інструментів для творчості та реалізації своїх ідей. Він дозволяє створювати різноманітні ігри та виробляти власні модифікації;
- інтеграція з Steam, будучи розроблений Valve, Source 2 має глибоку інтеграцію з платформою Steam, що полегшує розповсюдження та оновлення ігор, тобто , це приваблює більше для використання у розробці, тому що, можна виставити гру на платформі steam, для подальшого заробітку;
- ігрові ресурси та вміст, розробники можуть використовувати великий обсяг ігрових ресурсів та вмісту, доступних від Valve, для швидкого старту проектів;
- модульність та розширюваність, рушій побудований з урахуванням модульності, що дозволяє розробникам використовувати та розширювати його функціонал, додаючи власні розширення та інструменти.

Недоліки Source 2:

- обмежена поширеність , на відміну від інших популярних ігрових движків, таких як Unreal Engine чи Unity, Source 2 не так широко використовується в індустрії;

- відсутність підтримки macOS та Linux , поки що Source 2 не має повної підтримки для операційних систем macOS та Linux, що може бути обмеженням для розробників, які працюють у цих середовищах;
- більше підходить для внутрішньої розробки Valve , рушій Source 2 може бути оптимізований та орієнтований на внутрішні потреби Valve, що робить його менш універсальним для різних типів розробок.

Для встановлення Source 2 на комп'ютері, потрібно врахувати технічні

вимоги:

- система оперативної пам'яті (RAM)- мінімум 8 ГБ RAM, рекомендовано 16 ГБ або більше RAM;
- графічна карта (GPU) потрібна сучасна графічна карта з підтримкою DirectX 11 або новіше;
- місце на жорсткому диску , мінімум 20ГБ;
- процесор (CPU) мінімум двоядерний процесор з частотою 2,5 ГГц рекомендовано чотирьохядерний процесор з частотою 3,2 ГГц або вище;
- операційна система Windows 10 та вище.

1.4 Висновки до розділу

В цьому розділі було розглянуто засоби та технології створення ігор. В ході роботи було проведено аналіз існуючих мов програмування та було обрано наступні мови програмування JS, PHP, HTML, CSS, для створення бази даних – MySQL.

Після аналізу, вирішено створити гру для комп'ютера, з використанням сайту, перевага гри буде, в тому, що, буде база даних куди записуються дані про гравця та його обліковий запис, надалі, за допомогою цих даних, гравець може входити в свій акаунт і надалі вибирати ігри в які він гратиме.

2 РОЗРОБКА КОНЦЕПТ-ДОКУМЕНТУ ГРИ SNAKE

Концепт-документ – це текстовий або графічний документ, який висвітлює основну ідею та концепцію проекту. У сфері розробки ігор концепт-документ виступає важливим керівним матеріалом, що визначає основні аспекти гри перед початком активної розробки.

Елементи створення концепт документу:

- розробка сюжету;
- опис ідеї;
- мета та цільова аудиторія;
- сюжет гри;
- опис ігроладу;
- жанр гри;
- терміни виконання та бюджет;
- майбутня перспектива.

2.1 Розробка сюжету

Гра, яка була реалізована в цій кваліфікаційній роботі – «Змійка», зазвичай є досить простою та мінімалістичною. У кулінарному королівстві, де живуть різні їстівні і страви, з'явилася чарівна змійка на ім'я Бургеронія. Вона вирішила об'єднати свою любов до бургерів і свою незграбну природу, щоб стати королевою смачних пригод.

Бургеронія має особливу здатність: кожен бургер, який вона з'їдає, дарує їй зріст власного тіла. Її мета - подорожувати по кулінарному королівстві, збирати найсмачніші бургери і вдосконалювати свої здібності.

Але на шляху Бургеронії стоять виклики, які вона повинна здолати, для проходження рівня. Гравець повинен допомагати Бургеронії уникати

перешкод, збирати смачні бургери та розвивати її унікальні кулінарні здібності.

2.2 Опис ідеї

Ідея полягає в тому, створення простої та легкою гри, з взаємодією з сервером, також, з використанням мережевих технологій .

2.3 Мета та цільова аудиторія

Гра призначена для дітей від 3+, а також і для дорослих, кожен вік може грати в неї.

Основною метою гравця є контроль невеликого об'єкта, який представляє собою змію, і розвивати його, їдаючи їжу, яка з'являється на ігровому полі. Одночасно гравець повинен уникати зіткнень зі своїм власним тілом та межами ігрового поля.

Ключові аспекти мети гри "Змійка" включають:

1. зростання;
2. уникання зіткнень;
3. виживання та набуття очок;
4. підвищення складності;
5. досягнення цільового розміру;
6. перемогти в грі, тобто, набрати найбільшу кількість очок, та стати лідером в полі гравців.

Загалом мета гри "Змійка" полягає в найбільш ефективному зібранні їжі та уникненні перешкод для подовження та виживання змії на якнайбільш тривалий час.

2.4 Розробка та опис ігроладу

Ігровий процес або геймплей - це термін, який вказує на характеристику та динаміку взаємодії гравця з грою. Опис геймплею охоплює різноманітні аспекти гри, такі як механіка, системи, завдання, рівні, виклики та інші елементи, які формують враження від взаємодії з ігровим середовищем.

Ігровий процес у грі "Змійка" - це динамічний та захоплюючий процес, який включає в себе кілька ключових аспектів, такі як, збір їжі для зростання, збільшення швидкості гри та уникнення перешкод. Гра має простий геймплей та легко зрозумілі правила та дії для всіх аудиторій, це робить гру доступною широкому колу гравців. Механіка гри "Змійка" базується на простих, але захоплюючих принципах:

- зростання персонажа, розміри основного персонажа постійно зростають або збільшуються при поїданні їжі, коли гравець грає;
- уникання зіткнень – під час гри, гравець повинен уникати стикання зі стінкою або власним тілом, якщо щось із цього, якщо, щось із цього трапиться, гравець програє та гра розпочнеться заново, сторінка перезавантажується автоматично;
- з'їсти врожай. Головний персонаж повинен їжу, щоб збільшувати свій розмір та набирати бали. Розмір збільшується на одну клітинку, з кожною з'їденою їжею, кількість балів збільшується на одну одиницю;
- фінал гри є досягненням мети, тобто, набити найбільшу кількість очок, та перемогти. Перемога означає, внесення до таблиці переможців та кількість набраних очок.

Механіка гри "Змійка" є досить простою, але водночас вона може бути дуже веселою та викликаючою. Гравцеві доведеться бути уважним та стратегічним, щоб уникати зіткнень та тривати якнайдовше.

2.5 Аналіз жанру гри

Жанр гри "Змійка" можна визначити як аркадний або класичний ігровий жанр. Головною метою гравця є контроль невеликого об'єкта (зазвичай змійки), який росте в довжину кожного разу, коли він споживає певний об'єкт у грі. Гравець повинен уникати зіткнень зі стінами гри та самим собою.

2.6 Висновки до розділу

В ході кваліфікаційної роботи в цьому розділі було розроблено та описано концепт-документ гри, опис та розробка сюжету гри, мета та цільова аудиторія, опис ігроладу, аналіз жанру гри, терміни виконання та бюджет, майбутня перспектива. Проект розрахований для створення сайту з власної базою даних, полем реєстрації, щоб людина могла входити в аккаунт і вибирати надалі у що буде грати. Гра призначена для дітей від 3+, а також і для дорослих, кожен вік може грати в неї.

Основною метою гравця є контроль невеликого об'єкта, який представляє собою змію, і розвивати його, їдаючи їжу, яка з'являється на ігровому полі. Одночасно гравець повинен уникати зіткнень зі своїм власним тілом та межами ігрового поля.

3 РОЗРОБКА ІГРОВОГО РІВНЯ ГРИ SNAKE

Для рішення поставленої задачі в ході роботи буде розроблено ігровий рівень SNAKE, з полем реєстрації гравця та збереженням даних в базі даних. В ході роботи будуть використовуватися такі мови програмування, як, JS, PHP, HTML, CSS, SQL.

3.1 Розробка узагальненого алгоритму ігрового рівня

Основна мета створення алгоритму, який буде керувати логікою та взаємодіяти з гравцем (рис. 3.1). Загальний опис алгоритму виглядає так:



Рисунок 3.1 –Алгоритм ігрового рівня

1. Основний цикл гри.
2. Стартові параметри.
3. Рух змійки.
4. Інтерфейс користувача.
5. Рахунок та відображення
6. Обробка клавіш та введення гравця.
7. Зростання змійки та збір їжі.
8. Загибель та завершення гри.
9. Збереження та завантаження.

Розглянемо ці пункти детальніше:

1. Запуск основного циклу гри, що буде виконуватися безкінечно протягом гри, допоки гравець, не завершить гру або не спрацює умова загибелі змійки. Також, Обробка введення гравця для зміни напрямку руху змійки.

2. Треба, визначитись с параметрами ігрового поля, та інших стартовів параметрів. Розмістити змійку на полі та встановлення початкового напрямку її руху.

3. Має працювати цикл та умова, які, будуть оновляти рух змійки та перевіряти на умову зіткнення зі стінками чи іншими частинами змійки.

4. Відображення інтерфейсу гравця, такого як рахунок, вказівки чи інша інформація. Можливість перезапуску гри чи виходу до головного меню.

5. Обчислення рахунку гравця на основі з'їденої їжі або інших критеріїв. Відображення рахунку та іншої інформації на екрані гри.

6. Визначення та обробка введення гравця, яке може змінювати напрямок руху. Зміна напрямку руху змійки відповідно до натискання відповідних клавіш.

7. Перевірка, чи змійка з'їла їжу (наприклад, морква або яблуко) на поточній позиції. Якщо, їжа з'їдена, збільшення довжини змійки та генерація нової їжі на ігровому полі;

8. Перевірка умови, за якої гравець програє (зіткнення зі стінками чи самою собою). Зупинка гри та відображення результатів (наприклад, рахунку).
9. Збереження результатів гри та можливо для таблиці рекордів.

3.2 Розробка локацій ігрового контенту

Змійка - класична гра, яку ми знаємо ще з давніх часів. Оскільки проект створено за допомогою HTML та CSS, та на чистому JS, після цього, буде створено базу даних MySQL та поле реєстрації та входу в акаунт користувача на мові PHP.

Для того щоб створити гру на чистому JS, потрібно використовувати технологію CANVAS,

Елемент `<canvas>` (рис 3.2) має лише два атрибути - ширину та висоту. Обидва вони не є обов'язковими і можуть бути виставлені з використанням властивостей DOM. Якщо атрибути висоти та ширини не встановлені, canvas буде за замовчуванням шириною 300 пікселів і заввишки 150 пікселів. Ви також можете виставити розміри довільно в CSS, але під час рендерингу зображення масштабуватиметься відповідно до його розміру та орієнтації. (рис 3.2).

```
<canvas id="tutorial" width="150" height="150"></canvas>
```

Рисунок 3.2 - Приклад тегу `<canvas>`

Для проекту знадобляться два зображення, перше буде ігрове поле (рис 3.3), друге буде їжа для змійки. Щоб додати ці об'єкти до гри, ми можемо використовувати Конструктор `Image()`. Він може створювати новий екземпляр `HTMLImageElement`. Еквівалентно `document.createElement('img')` (рис 3.4).

```
const ground = new Image(); // Створення об'єкта поля
ground.src = "img/ground.png"; // Вказуємо потрібне зображення поля

const foodImg = new Image(); // Створення об'єкта їжі
foodImg.src = "img/food.png"; // Вказуємо потрібне зображення їжі
```

Рисунок 3.3 - Приклад створення та підключення об'єкта через конструктор Image()

Поле буде розміром 608 на 608 пікселей. , під час гри, гравець буде грати на світло зелених і темно зелених клітинах, як на шаховій дошці. За полем, буде межі, які не можна перетинати (рис 3.4)

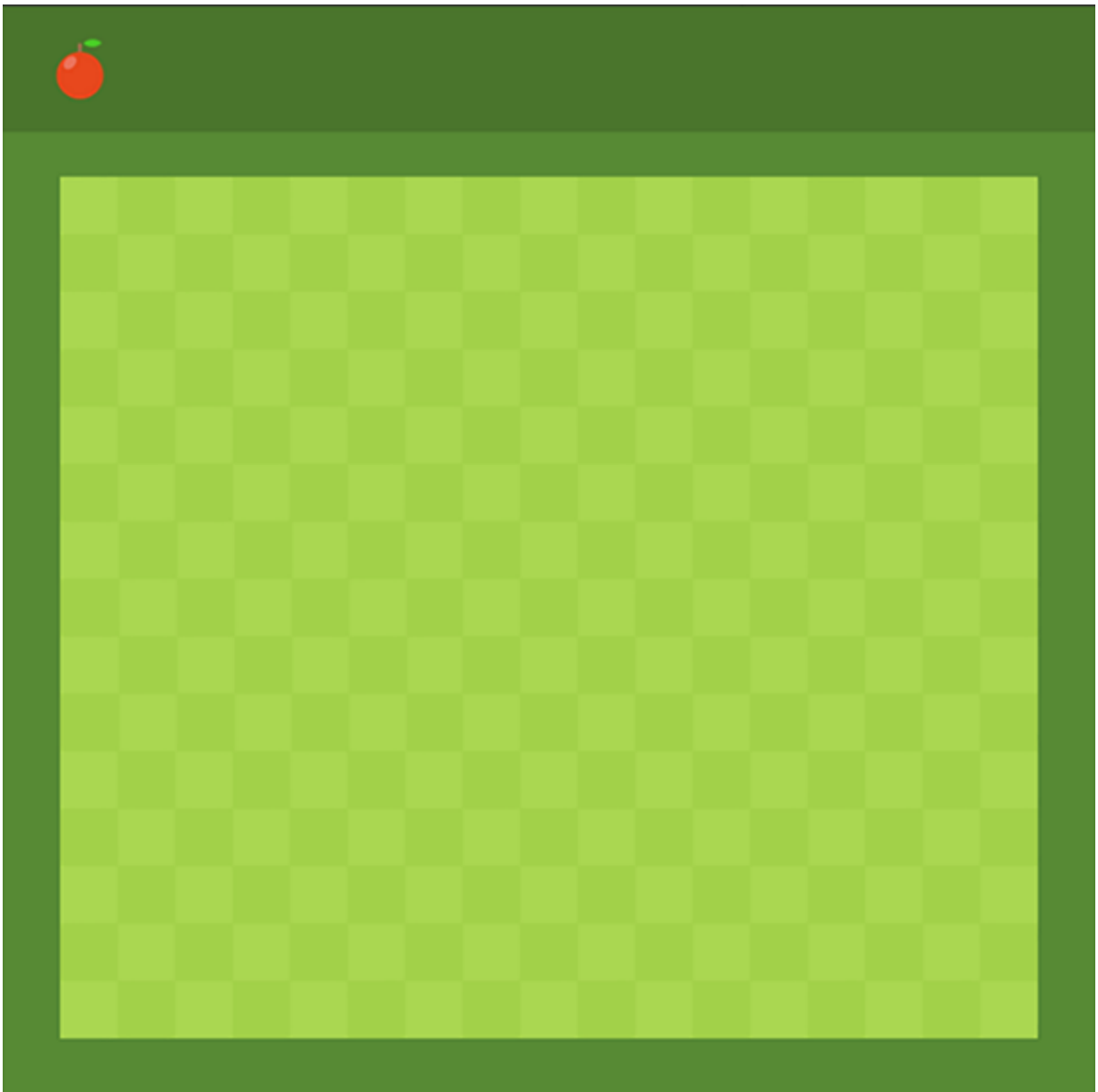


Рисунок 3.4 – Ігрове поле гри

3.3 Етапи розробки проекту

Для розробки сайту з грою, знадобиться створити проект. Для цього, потрібно створити папку з назвою snake. (рис 3.5)



Рисунок 3.5 - Створення папки Snake

Всередині папки snake створено папку www це надалі допоможе мені щоб до локальної бази даних звертатися.

Для початку, потрібно створити чистий проект на HTML та CSS всередині папки www. Кожен класичний сайт чи проект у браузері створюється так, як у рис 3.6.



Рисунок 3.6 - Створення папки www

В кожному проекті створюється папка css, js та img. В папці css зберігаються файли з розширенням css в них розписують зовнішній вид сайту, об'єктів, тощо. В папці js зберігаються файли з розширенням js, в них які містять вихідний код, написаний на мові програмування JavaScript для створення динамічного контенту на веб-сайтах, включаючи взаємодію з користувачем, маніпуляції DOM (об'єктний представник документа) та інші функції. Надалі, буде створювати та додавати файли в цю папку, щоб працювати з ними. (рис 3.7)

css	05.01.2023 0:04	Папка с файлами	
img	08.01.2023 14:42	Папка с файлами	
js	13.12.2023 22:22	Папка с файлами	
index	05.01.2023 13:50	Opera Web Docu...	8 КБ

Рисунок 3.7- Класична папка з проектом

Далі потрібно створити HTML сторінку яка може містити класичну розмітку, в яку необхідно вписати CANVAS для відображення гри всередині нього (рис 3.8).

Приклад коду:

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <link rel="stylesheet" href="css/style.css">
7    <title>Snake classic</title>
8  </head>
9  <body>
10   <canvas id="game" width="608" height="608"></canvas>
11   <script src="js/game.js"></script>
12 </body>
13 </html>

```

Рисунок 3.8 - Приклад створення основного файлу та тег CANVAS

Зараз, потрібно створити в папці js два файли game.js та password.js. В папці css, створив файл style.css, де буду записувати всі стилі та зовнішній вигляд, рис. 3.9. та рис 3.10.

game	13.12.2023 21:36	Исходный файл J...	3 КБ
password	13.12.2023 22:34	Исходный файл J...	1 КБ

Рисунок 3.9 - Створення файлів game.js та password.js.

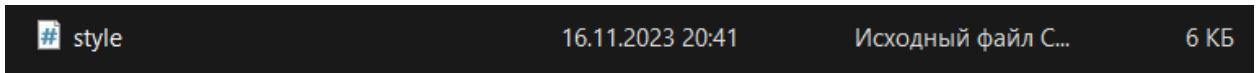


Рисунок 3.10 - Створення файлу style.css.

Для початку, відкривав все те що створив в VISUAL STUDIO CODE (VSC) (рис 3.11).

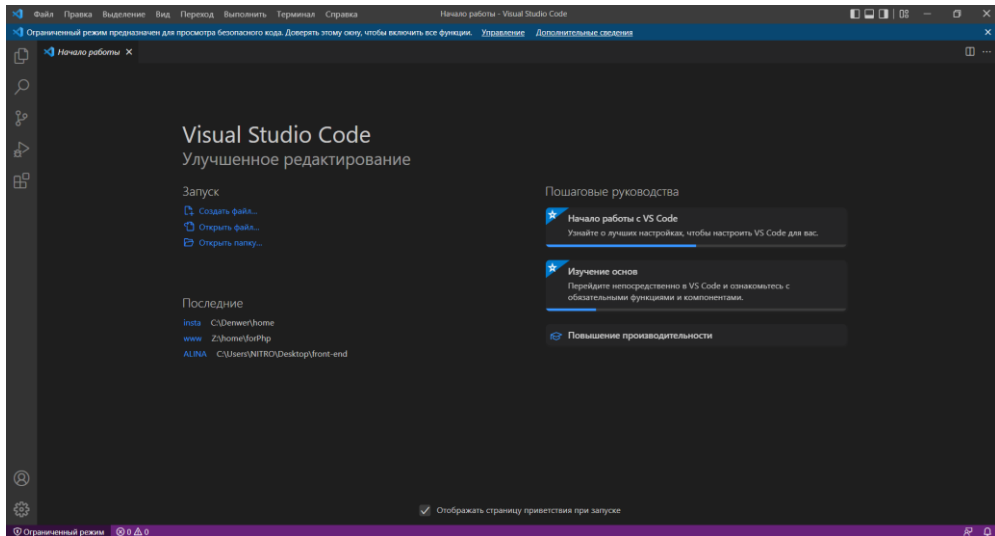


Рисунок 3.11 – Початок роботи на Visual Studio Code

Після цього заходжу в VSC та в ново створенню папку (рис 3.12 та рис 3.13).

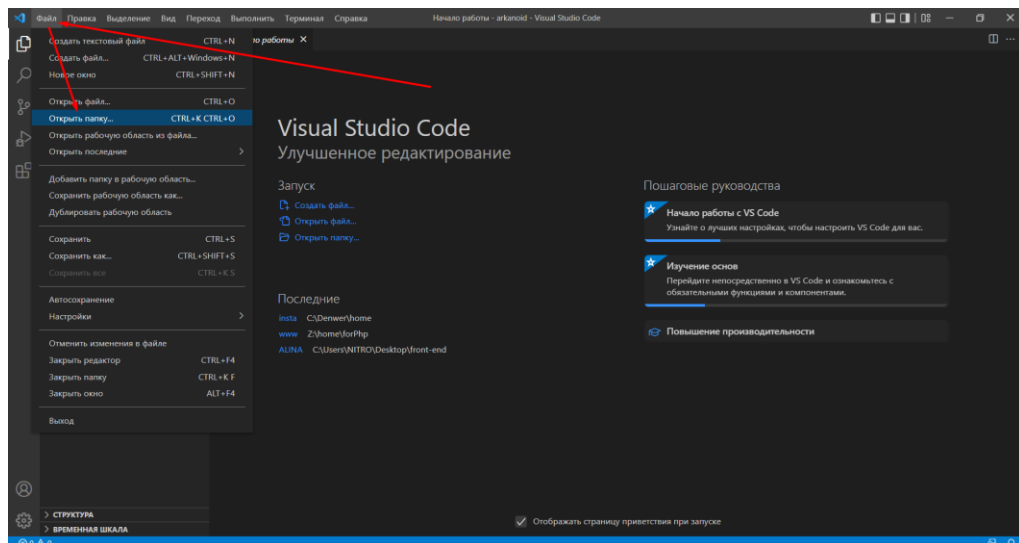


Рисунок 3.12 – Відкриття папки snake в VSC

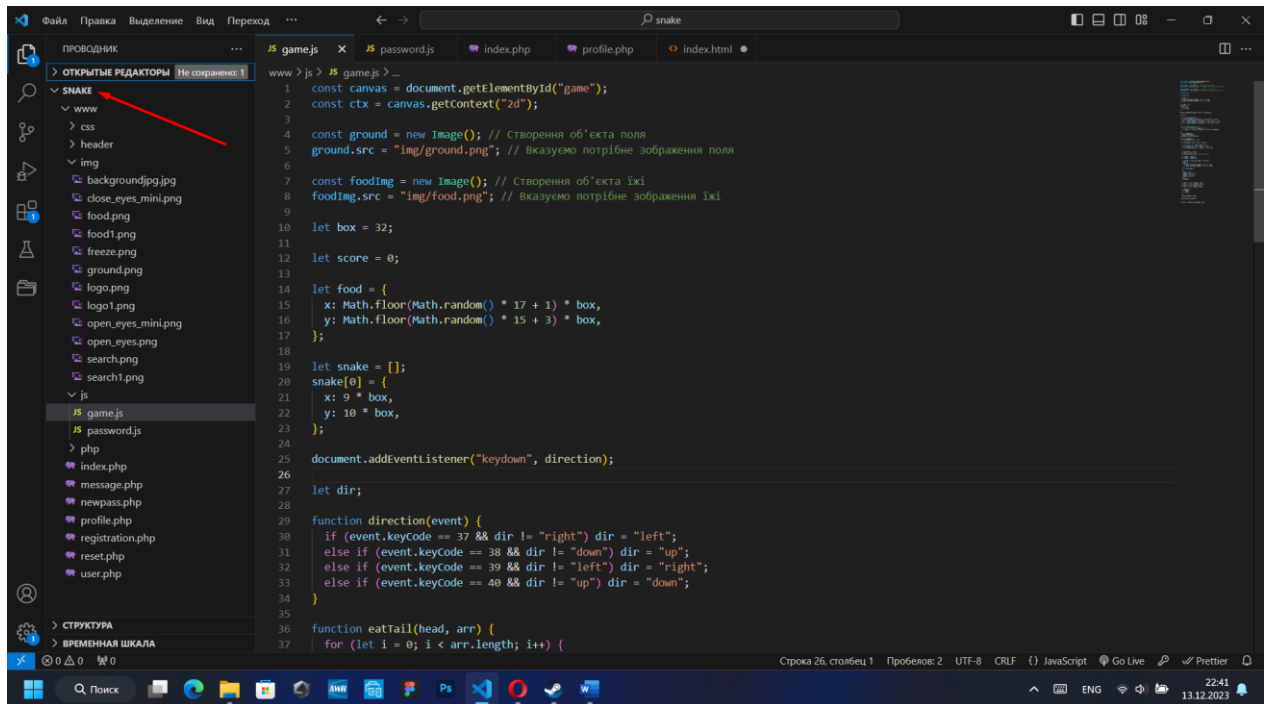


Рисунок 3.13 – Відкриття папки snake в VSC для подальшої роботи

В VSC відкривається папка game.js та починаю писати власну гру. Для власної гри потрібно створити 4 змінні (рис 3.14 та рис 3.15), перша це змійку, потім їжу та та лічильник, також, розмір кожної клітки 32. У JavaScript файлу треба додати вибірку тегу <canvas>, а також треба вказати контекст гри, у нас вона “2D”.

```

let box = 32;

let score = 0;

let food = {
  x: Math.floor(Math.random() * 17 + 1) * box,
  y: Math.floor(Math.random() * 15 + 3) * box,
};

let snake = [];
snake[0] = {
  x: 9 * box,
  y: 10 * box,
};

```

Рисунок 3.14 – Створення змінних

```
const canvas = document.getElementById("game");
const ctx = canvas.getContext("2d");
```

Рисунок 3.15 – Створення змінних

Food об'єкт кожного разу коли змія буде з'їдати її, за допомогою Math.random() їжа буде опинятися в різних місцях. Для відтворення об'єктів в проєкті створено змінні ground та food, як на рис. 3.2. Для того щоб, комп'ютер розумів, куди моя змійка буде направлятись додано до документу обробник подій addEventListener (рис 3.16).

```
document.addEventListener("keydown", direction);
```

Рисунок 3.16 – Створення обробнику подій addEventListener

При кожному натискання клавiш стрiлочок напрямку, змійка рухатиметься за певним напрямком, та для неї створено функцію direction, яка відповідає за ці події (рис 3.17).

```
function direction(event) {
  if (event.keyCode == 37 && dir != "right") dir = "left";
  else if (event.keyCode == 38 && dir != "down") dir = "up";
  else if (event.keyCode == 39 && dir != "left") dir = "right";
  else if (event.keyCode == 40 && dir != "up") dir = "down";
}
```

Рисунок 3.17 – Створення обробнику подій addEventListener

Це перевірка умови, перевіряє, якщо змійка з'їла їжу, збільшує лічильник скільки з'їдено і додає на випадковому місці їжу (рис 3.18).

```

if (snakeX == food.x && snakeY == food.y) {
  score++;
  food = {
    x: Math.floor(Math.random() * 17 + 1) * box,
    y: Math.floor(Math.random() * 15 + 3) * box,
  };
} else snake.pop();

```

Рисунок 3.18 – Створення умови if для перевірки їжі

Це цикл for для додавання довжини змійки, голова у моєї змійки буде зелена, все інше тіло буде червоним (рис 3.19).

```

for (let i = 0; i < snake.length; i++) {
  ctx.fillStyle = i == 0 ? "green" : "red";
  ctx.fillRect(snake[i].x, snake[i].y, box, box);
}

```

Рисунок 3.19 – Створення умови if для перевірки їжі

Щоб функція працювала постійно, необхідно запустити її виконання через setInterval() (рис 3.20).

```

let game = setInterval(drawGame, 100);

```

Рисунок 3.20 – Створення setInterval

Функція закінчення гри, після цього викликаю функцію clearInterval(), та перезавантажую сторінку. (рис 3.21)

```

function endgame() {
  clearInterval(game);
  document.location.reload();
}

```

Рисунок 3.21 – функція endgame()

Ось так виглядає моя папка з моїм проектом саме зараз (рис 3.22)

css	16.11.2023 18:36	Папка с файлами	
header	16.11.2023 18:36	Папка с файлами	
img	13.12.2023 11:23	Папка с файлами	
js	16.11.2023 20:07	Папка с файлами	
php	16.11.2023 18:36	Папка с файлами	
index	29.11.2023 9:22	Исходный файл Р...	3 КБ
message	16.11.2023 18:46	Исходный файл Р...	2 КБ
newpass	16.11.2023 18:46	Исходный файл Р...	2 КБ
profile	13.12.2023 21:38	Исходный файл Р...	3 КБ
registration	16.11.2023 19:58	Исходный файл Р...	3 КБ
reset	16.11.2023 18:46	Исходный файл Р...	3 КБ
user	16.11.2023 18:46	Исходный файл Р...	1 КБ

Рисунок 3.22 - Приклад моєї папки з моїм проектом

Далі потрібно додати реєстрацію, створення акаунту, відновлення пароля, внести все в базу даних, зробити відповідне меню користувача і т.і.

3.4 Висновки до розділу

В цьому розділі, були детально описані кроки створення гри змійки. Було розроблено алгоритм гри. Основна мета створення алгоритму, це описання процесу керування логікою та взаємодія з гравцем.

Також було створено локації ігрового контенту. Для того щоб створити гру на чистому JS, використовували технологію CANVAS. Елемент `<canvas>` має лише два атрибути - ширину та висоту, причому обидва вони не є обов'язковими і можуть бути встановлені за допомогою властивостей DOM. У випадку, якщо атрибути висоти та ширини не задані, елемент `<canvas>` автоматично приймає значення за замовчуванням - ширина 300 пікселів і висота 150 пікселів.

Описані етапи створення проекту, Для розробки сайту з грою, було створено папку з назвою snake. В папці snake помістилися такі папки , як , css , header, img , js, php. В папці css, зберігаються файл зі стилями, в папці js файл з самою грою змійка , img , різні фотографії для сайту. В папці php та header , по проекту будуть надалі використовуватися.

Для створення гри, потрібні були знання мові програмування JS , де використовувались такі знання: цикли for , умови if, різні function , обробнику подій addEventListener.

4 РОЗРОБКА ТА ОРГАНІЗАЦІЯ ВЗАЄМОДІЇ ІІ З СЕРВЕРОМ ONLINE ГРИ

Для розробки сайту та організація взаємодії її з сервером online гри знадобиться серверна мова програмування PHP.

PHP - це популярна мова програмування, особливо серед веб-розробників. Автором початкової версії є Расмус Лердорф, ідея якого полягала у розробці набору інструментів спрощення процесу створення динамічних веб-сторінок. Незважаючи на те, що сучасний PHP є мовою загального призначення, найчастіше його використовують як серверний інструмент для генерації HTML-коду, який потім інтерпретується веб-браузером. В моєму проєкті PHP знадобиться для того, щоб гравці могли зареєструватись.

Для того, щоб створити форму реєстрації, знадобиться мова PHP, особливо цієї мови програмування входять:

- легкий синтаксис;
- найголовнішою особливістю є захищеність даних, тобто повністю прихований код від користувачів сайту;
- відмінна сумісність і переносимість - PHP-коди працюють однаково добре з різними платформами;
- висока швидкість роботи та, відповідно, загальна продуктивність ресурсів.

Щоб користуватись мовою PHP, потрібен буде локальний сервер. В особливості створення проєкту, лежить створення локального сервера, а тому, функція Денвер дуже гарно може допомогти вирішити цю проблему.

У базовий пакет входять такі програми як:

- веб-сервер Apache з підтримкою SSI, SSL, mod_rewrite, mod_php.
- Інтерпретатор PHP з підтримкою GD, MySQL, SQLite;
- в СУБД MySQL з підтримкою транзакцій (mysqld-max);
- панель phpMyAdmin для адміністрування СУБД;

- ядро інтерпретатора Perl без стандартних бібліотек (поставляються окремо). Емулятор sendmail та сервера SMTP з підтримкою роботи спільно з PHP, Perl, Parser.

4.1 Розробка серверної частини

Для початку потрібно створити основний файл для роботи, це `index.php`.

(рис 4.1).

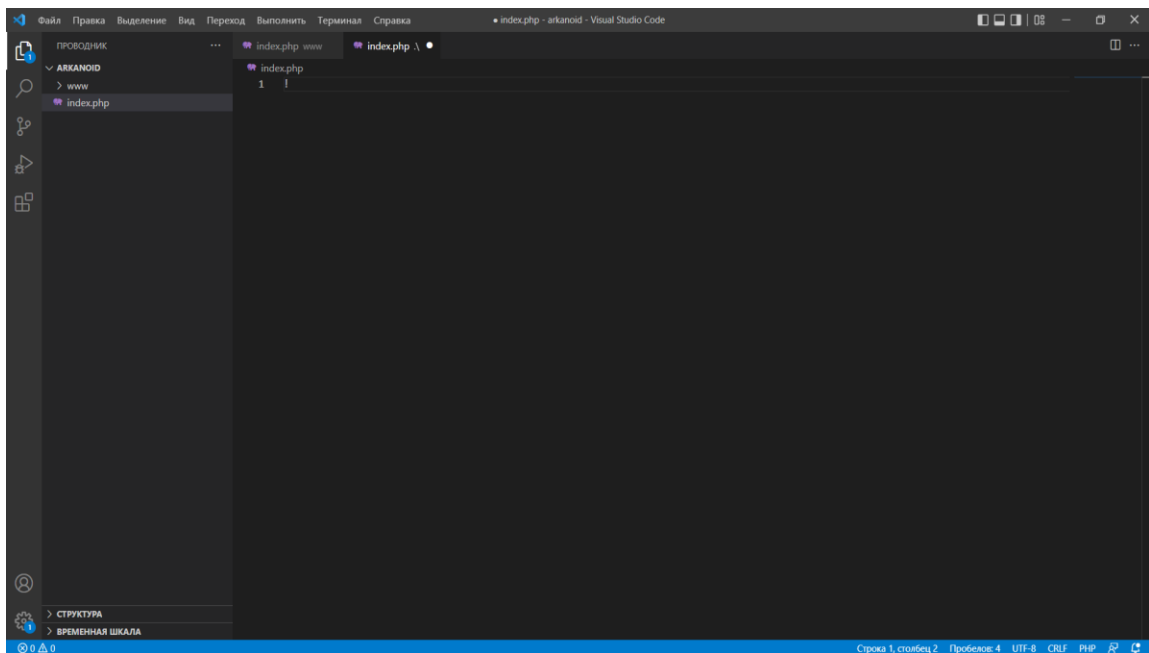


Рисунок 4.1– Index.php в VSC

Після цього, потрібно створити стандартний шаблон HTML. В теґі `<body>`, створюємо форму реєстрації гравця, щоб грати в цю гру, кожен гравець потрібен зареєструватися. Це робить проект унікальним та забезпечує безпеку даних гравців (рис 4.2).

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10
11 </body>
12 </html>

```

Рисунок 4.2– HTML шаблон в index.php

PHP код пишеться в спеціальних символах `<?php ... ?>` треба всередині писати код. Особливість серверної мови в тому, що кожна змінна починається зі знака `$`. Кожна змінна, може приймати будь який тип, наприклад, в мові програмування JAVA або C#, змінні створюють заздалегідь, і, змінна, з іменем `name` з типом `STRING`, не може приймати числові данні, тільки символні та строчні, а в PHP змінна `$name` може в одній строчці коду, бути зі значенням `STRING`, а в іншій часті коду, буде чисельне значення.

```

1 <?php
2 session_start();
3 require_once 'connect.php';
4
5 $login = $_POST['login'];
6 $password = md5($_POST['password']);
7 $_SESSION['login'] = $login;
8 $_SESSION['password'] = $password;
9 $check_login = mysqli_query($mysqli, "SELECT * FROM users WHERE `login` = '$login' AND `password` = '$password'");
10 $check_log = mysqli_query($mysqli, "SELECT * FROM users WHERE `login` = '$login'");
11
12 if (mysqli_num_rows($check_login) == 1) {
13     $user = mysqli_fetch_assoc($check_login);
14     $_SESSION['id_profile'] = $user['id'];
15     $_SESSION['phone_profile'] = $user['mobile_phone'];
16     $_SESSION['name_profile'] = $user['name'];
17     $_SESSION['login_profile'] = $user['login'];
18     header('location: ../profile.php');
19 } else if (mysqli_num_rows($check_log) == 1) {
20     $_SESSION['message'] = "Mistake in password, please, try again write password";
21     unset($_SESSION['login']);
22     unset($_SESSION['password']);
23     header('location: ../index.php');
24 } else {
25     $_SESSION['message'] = "WE DON'T FOUND LOGIN OR PASSWORD, PLEASE , CHECK YOUR LOGIN OR PASSWORD ONE MORE ";
26     unset($_SESSION['login']);
27     unset($_SESSION['password']);
28     header('location: ../index.php');
29 }
30 ?>

```

Рисунок 4.3 – Приклад коду з PHP файла

Для написання форми, знадобиться тег `<form>` і 2 теги `<input>` з кнопкою `<button>`, особливість написання форми лежить у php, за допомогою мови програмування php потрібно відправляти дані на сервер, перевіряти, чи є в базі така людина і якщо, людина знайдена, після цього дозволити заходити їй на сайт. В самому теґі form, є атрибути, action, method, атрибут action, потрібен для того, щоб визначити на якій сторінці будуть у нас дії, якщо цей атрибут залишити порожнім, компілятор буде шукати код по даній сторінці. Також, атрибут method, існують два методи GET та POST. Основна відмінність методу GET від POST у способі передачі. Запит GET передає дані в URL у вигляді пар "ім'я-значення" (іншими словами через посилання), а запит POST передає дані в тілі запиту (рис 4.4).

```
<form action="php/in.php" method="post" class="form2">
  <input type="text" name="login" placeholder="Login" value="<?= $_SESSION["login"] ?>" required>
  <input type="password" name="password" id="pass" placeholder="Password" value="<?= $_SESSION["password"] ?>"
  <span id="icon" class="icon1"></span>
  <button type="submit" class="sub1" name="inp">Input</button>
</form>
```

Рисунок 4.4 – Приклад написання теґа `<form>`

Для проекту знадобляться знання про сесії, масиви GET/POST, щоб отримувати дані з таблиць, підключення до БД та звичайні умови. Вся суть php буде в полі реєстрації, отримувати через тег `input` значення (ім'я, номер телефону, логін, пароль), (рис. 4.5), записує все в базу даних, після успішної реєстрації, ви направитеся на сторінку входу в обліковий запис, також треба ввести логін, пароль, якщо значення всі збіглися (рисунок 4.6), ви потрапите на сторінку гри. Після цього, створюю ще 2 сторінки, відновлення пароля та підтвердження особи, що саме та людина, хоче поміняти пароль в акаунті.

The screenshot shows a web browser window with the address bar displaying 'snake/registration.php'. The main content area contains a registration form titled 'SNAKE'. The form consists of four input fields: 'Mobile', 'Full name', 'Login', and 'Password'. Below these fields is a blue button labeled 'Registration'. Underneath the button is the text 'OR' and a link that says 'Forgot your password?'. At the bottom of the form is a button that says 'Do you have account? Input'. An arrow from the right points to the form with the label 'ПОЛЕ РЕЄСТРАЦІЇ ГРАВЦЯ'.

Рисунок 4.5 – Форма реєстрації

The screenshot shows a web browser window with the address bar displaying 'snake/index.php'. The main content area contains a login form titled 'Snake'. The form consists of two input fields: 'Login' and 'Password'. Below these fields is a blue button labeled 'Input'. Underneath the button is the text 'OR' and a link that says 'Forgot your password?'. At the bottom of the form is a button that says 'Do you not have account? Registration'. An arrow from the right points to the form with the label 'ПОЛЕ ДЛЯ ВХОДУ В АКАУНТ'.

Рисунок 4.6 – Форма для входу в акаунт гравця

Це була форма реєстрації та входу в акаунт користувача, далі розглянемо розділ створення бази даних MySQL. в MySQL буде створена таблиця даних, куди зберігатиму всі дані про користувачів, їх паролі, дати створення, логіни та паролі гравців.

4.2 Розробка бази даних

База даних у проєкті використовується для зберігання інформації про гравців, їхні набрані бали та для відновлення облікового запису. Для створення цієї бази даних використовується програма phpMyAdmin, яка входить у базовий пакет Denwer. Для доступу до неї потрібно в адресному рядку будь-якого браузера ввести localhost/tools/phpmyadmin/, як показано на рис. 4.7.

localhost/tools/phpmyadmin/

Рисунок 4.7 – Адресний рядок в браузері localhost/tools/phpmyadmin/

Ця адресна строка переведе до користувача phpMyAdmin, в якому буде створюватись базу даних, рис. 4.8

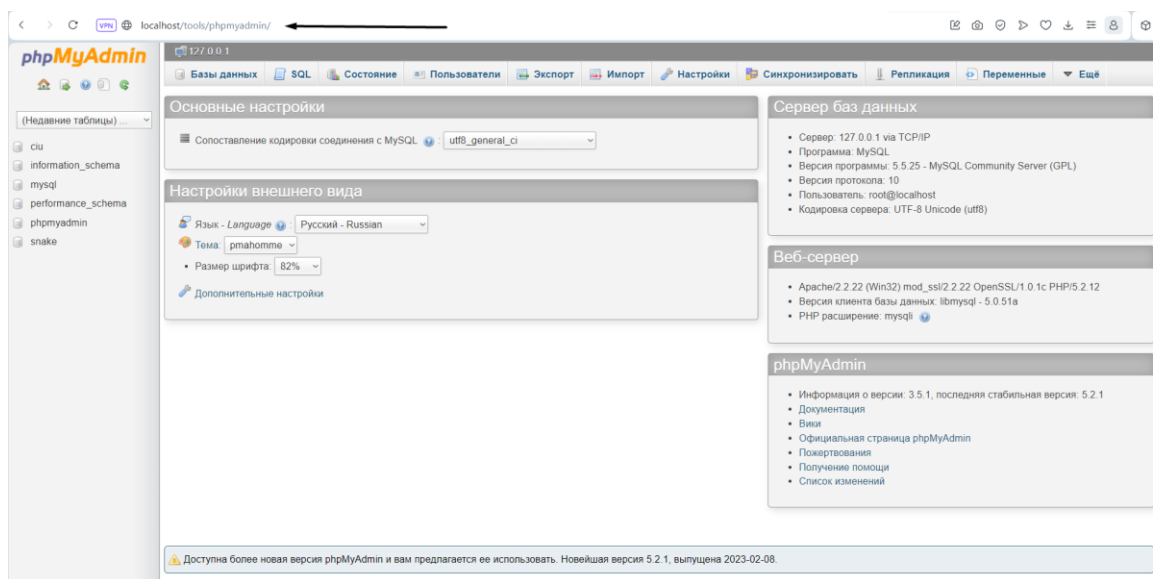


Рисунок 4.8 – Меню користувача phpMyAdmin

Щоб створити базу даних, потрібно перейти в розділ Бази даних, ввести назву своєї бази даних, обрати в полі порівняння utf8_general_ci (рис 4.9 та рис 4.10).

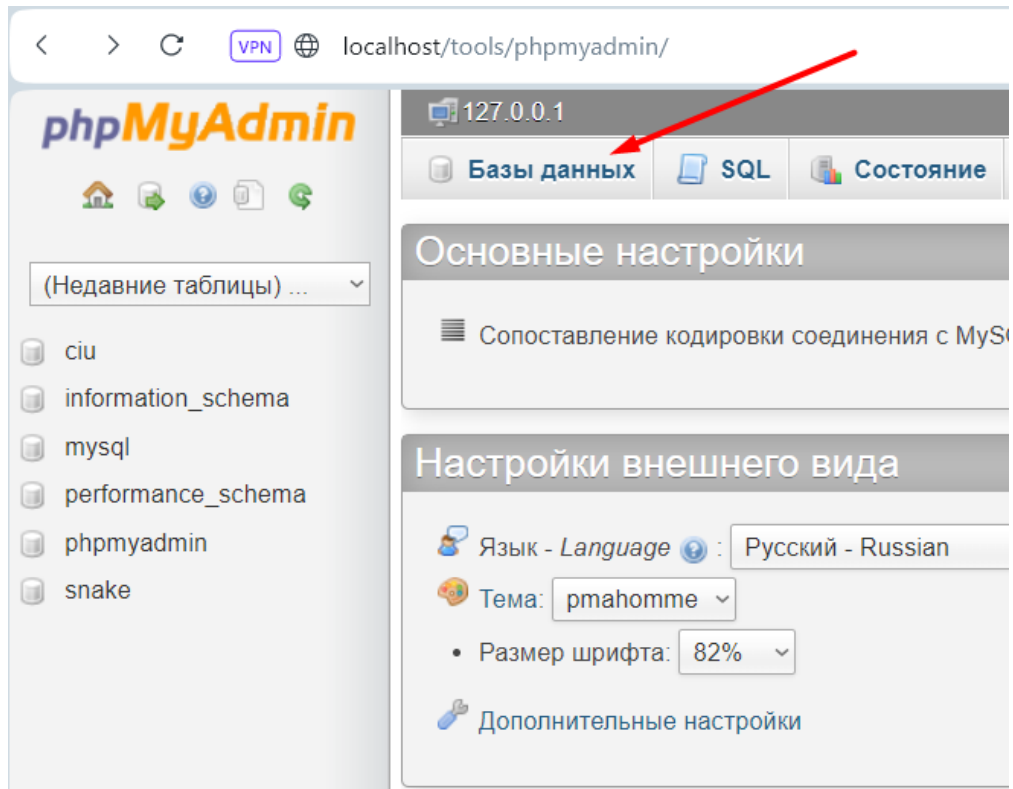


Рисунок 4.9 – Меню Базы данных для створення БД

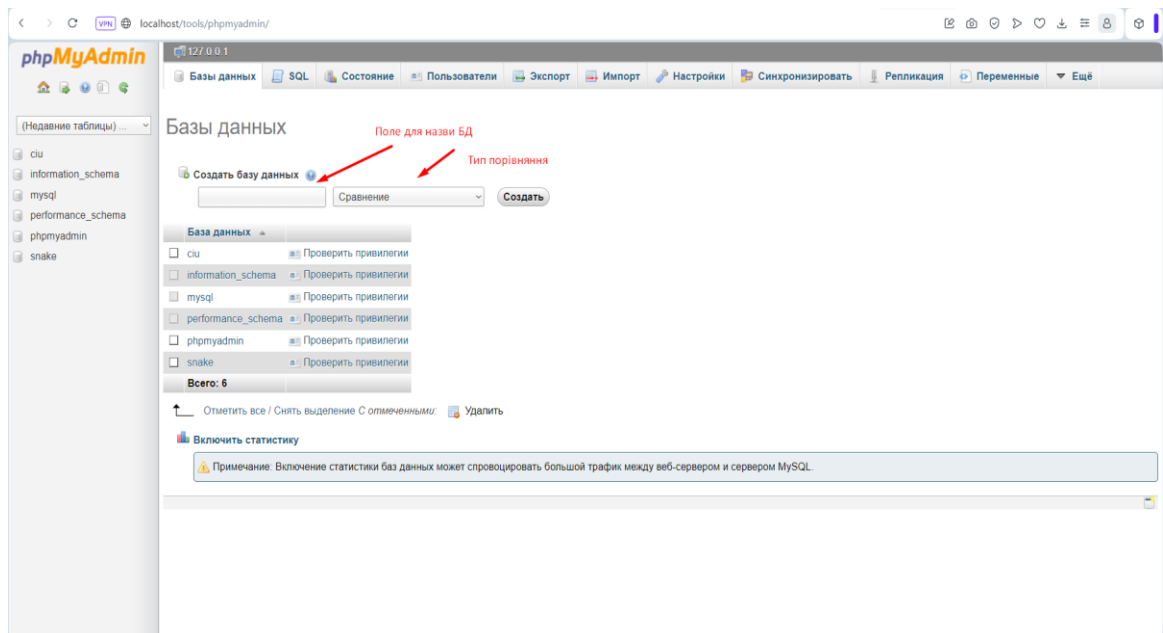


Рисунок 4.10 – Поле створення БД

Щоб створити таблицю, потрібно написати ім'я таблиці та кількість стовпців у даній таблиці буде створюватись таблиця з назвою users і для нашої таблиці знадобиться 7 стовпців, рис. 4.11.

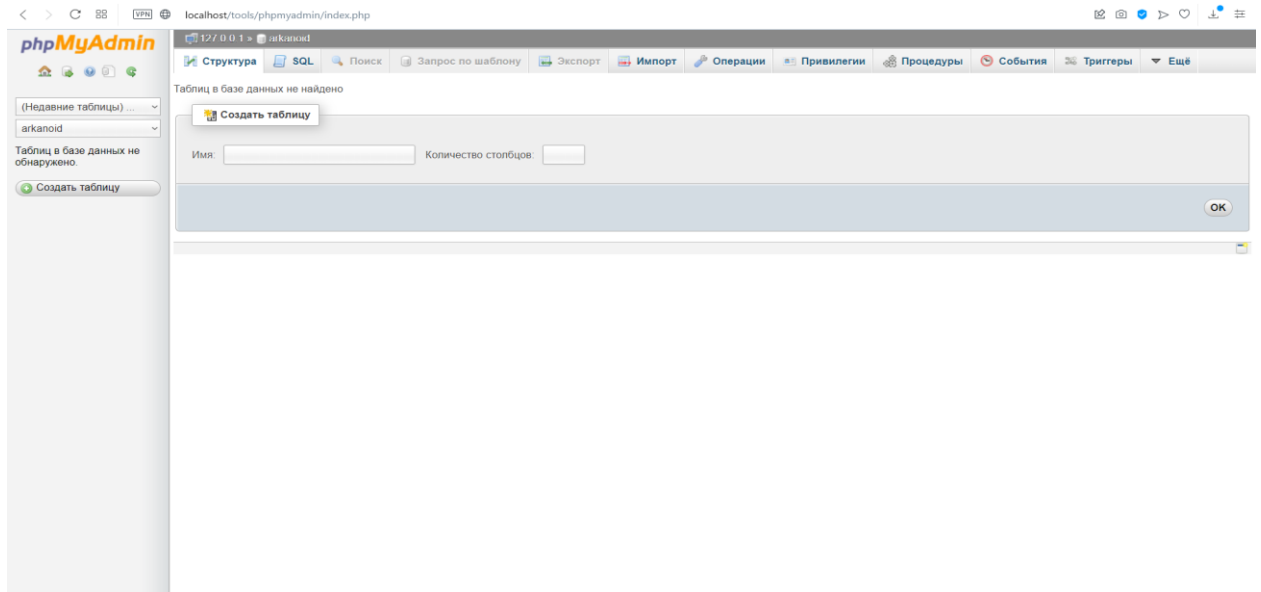


Рисунок 4.11 – Створення таблиці users

Для створення таблиці знадобиться, 7 колонок, “id”, “mobile_phone”, “name”, “login”, “password” та “change_key”, “game_points”.

У кожній з них мені необхідно визначити тип. Для колонки "id" та “game_points” використовую тип INT (цілі числа), а для всіх інших можна використовувати тип VARCHAR (рядки). Це пов'язано з тим, що ім'я, логін та пароль можна представити лише символами. Рисунок 4.12-4.13

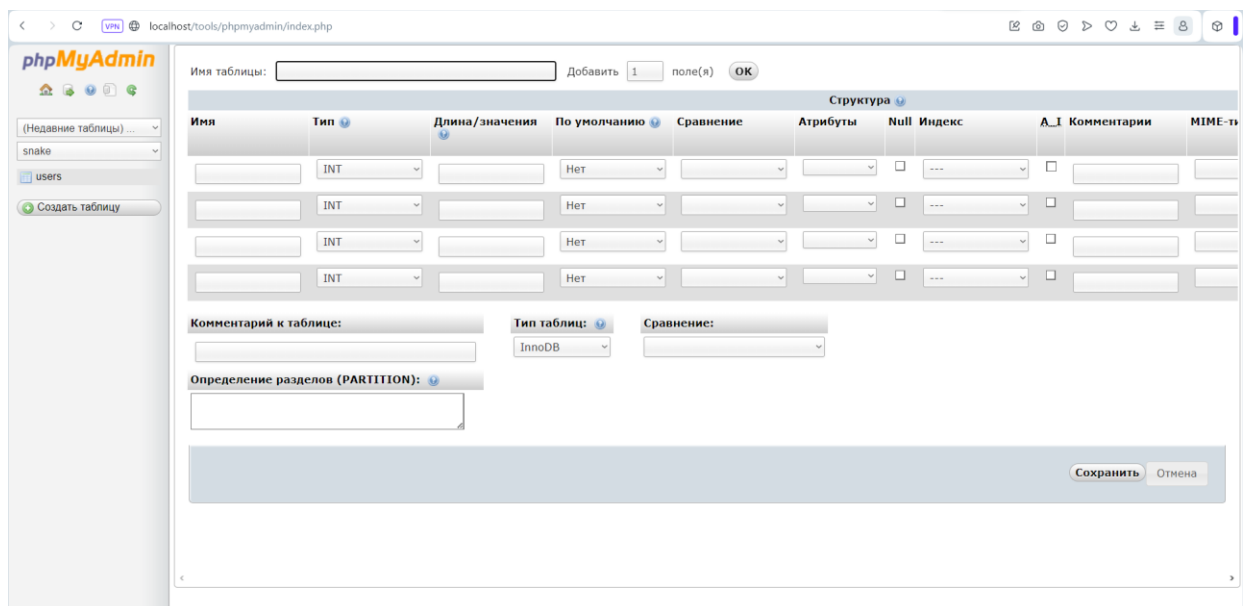


Рисунок 4.12 – Меню інтерфейса створення таблиці

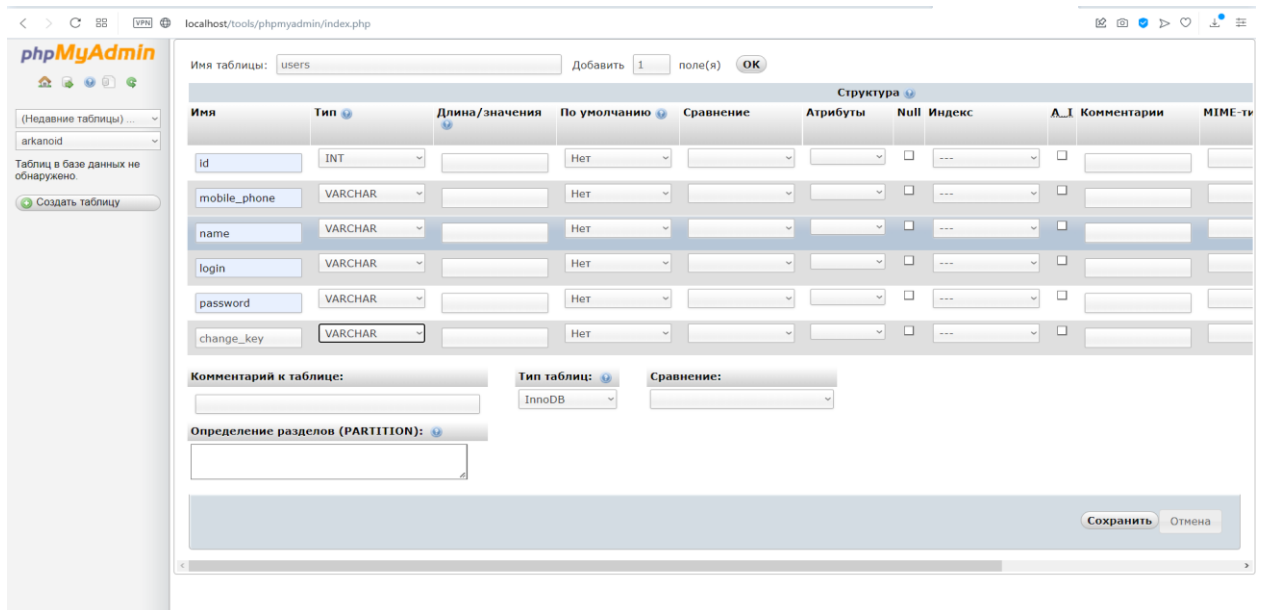


Рисунок 4.13 – Створення таблиці users

Тепер, для кожної колонки потрібно вказати, довжину, значень, тобто, якщо це символи, максимальна кількість допустимих символів, якщо це число, максимальне допустима значення. Для пароля поставлено значення 50, для всіх інших символів, поставлено 255.

Для типів VARCHAR, треба обрати порівняння utf8_general_ci, гравці можуть бути з різних країн і вказувати, наприклад, кирилицю або японські або китайські символи.

Для поле "id" встановлено атрибуту "unsigned" та індекс "primary" (первинний ключ), щоб спростити та прискорити пошук інформації про гравця в базі даних. Крім того, для полів "mobile_phone" та "login" встановлено атрибуту "unique" (унікальний). Логін не може повторюватися у двох чи більше гравців, а також номери мобільних телефонів мають бути унікальними.

Для типу id, встановив Auto_increment (A_I), для того щоб, при створенні нових користувачів, id збільшувалося на одиницю.

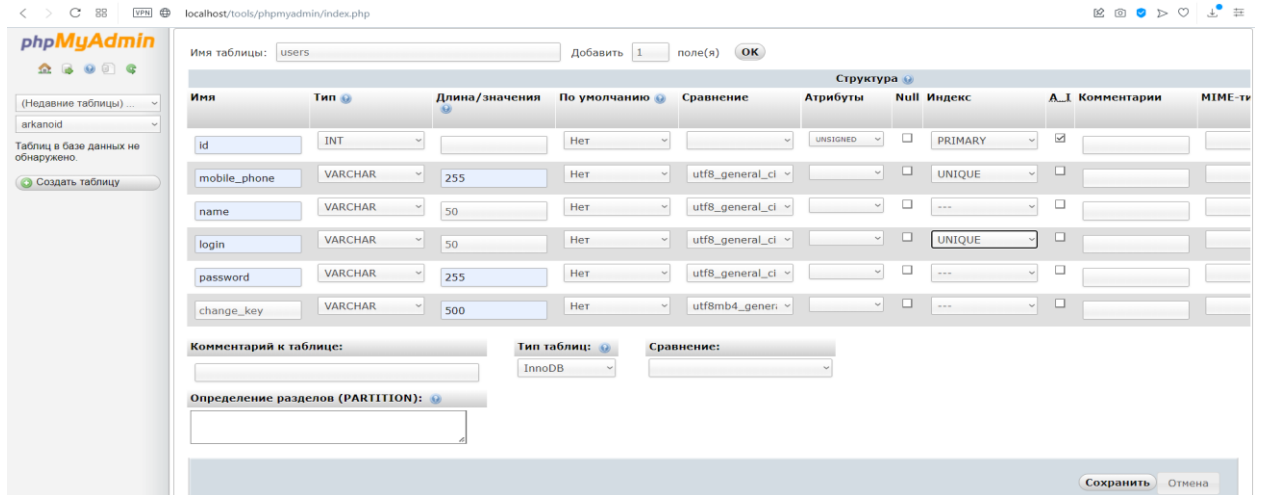


Рисунок 4.14 – Заповнення таблиці users

Тип таблиці, який обрано, - MyISAM, оскільки він має вбудований повнотекстовий пошук, на відміну від InnoDB. У відміну від MyISAM, InnoDB підтримує транзакції. Також InnoDB дозволяє використовувати блокування на рівні рядка, тоді як MyISAM використовує блокування на рівні таблиці. У InnoDB є можливість встановлювати обмеження зовнішніх ключів, що в MyISAM недоступно, і для порівняння використовується utf8_general_ci для всіх мов світу. Після цього ми натискатимемо кнопку "Зберегти" для збереження нашої таблиці.

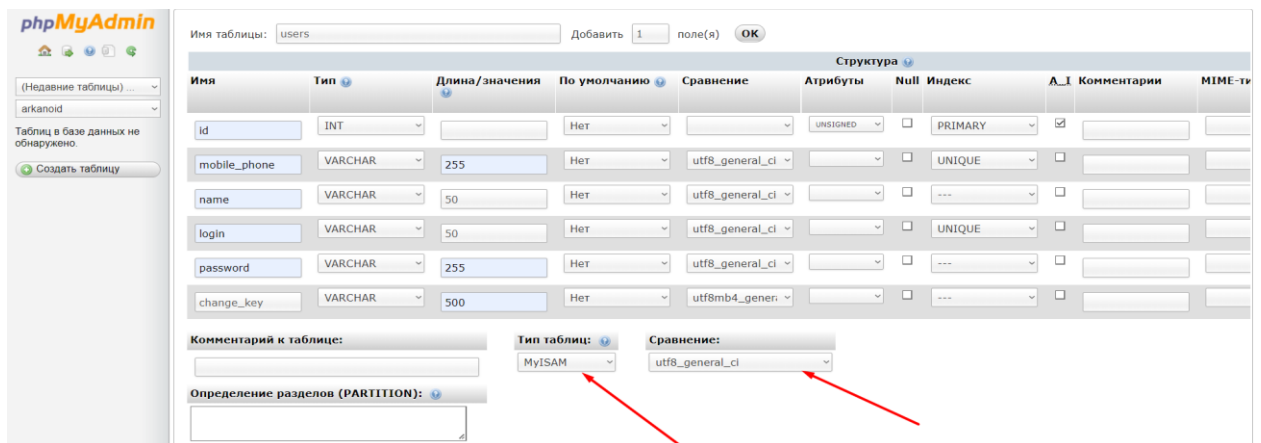


Рисунок 4.15 – Тип та порівняння у таблиці users

Після перевірки всіх даних збережено на рис. 4.16 буде показано, що база створена.

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> users	Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8_general_ci	32 КБ	-
1 таблица	Всего	2	InnoDB	utf8_general_ci	32 КБ	0 Байт

Рисунок 4.16 – Ново створена таблиця users

Тут можна створювати SQL-код напряму. Наприклад, якщо вам потрібно вибрати якісь дані з таблиці користувачів, просто напишіть запит: `SELECT * FROM USERS` (рис 4.17). Результат цього коду буде відображений нижче

The screenshot shows the phpMyAdmin interface with a SQL query executed. The query is `SELECT * FROM USERS LIMIT 0, 30`. The result is displayed in a table with the following columns: `id`, `mobile_phone`, `name`, `login`, `password`, `change_key`, and `game_points`. Two rows of data are shown, with red arrows pointing to the `id` and `mobile_phone` columns.

id	mobile_phone	name	login	password	change_key	game_points
21	380636755009	Bohdan Ponomarenko	bohonoma	00a1f187721c63501356b7f91e69382c	d41d8ccd98100b204e9800098ec8f8427e1576	NULL
22	380636744134	Maksim Taran	Maks1234	25d55ad263aa400af464c76d713c07ad	d41d8ccd98100b204e9800098ec8f8427e1576	NULL

Рисунок 4.17 – SQL запити в phpMyAdmin

Код PHP який підключає ново створену базу до проекту (рис 4.18).

```
<?php
$mysqli= mysqli_connect("localhost","root","","snake");
$mysqli->query("SET NAMES 'utf8'");
if (!$mysqli) {
die('Error connect to DB');
}
?>
```

Рисунок 4.18 – Підключення БД до проекту

4.3 Аутентифікація та безпека

За аутентифікацію та безпеку відповідає файл `in.php`, `change.php`, та `regist.php`. `Regist` та `In` це відповідно, реєстрація та вхід до облікового запису,

скрізь перевіряють введені дані, якщо дані для реєстрації вперше введені, обліковий запис створюється, якщо ні, просять заново ввести дані які реально існують і не зайняті. (рис 4.19).

Enter your email address, username or phone number and we will send you a link to restore access to your account.

Mobile or Login

E-mail

Change password

OR

[Create a new account?](#)

[Do you have account? Input](#)

Рисунок 4.19 – Сторінка скидання паролю

При зміні пароля користувача спершу проводиться перевірка інформації, на яку пошту або номер телефону було зареєстровано дані. Якщо будь-яка з цих інформацій знаходиться, користувачу висилають дані на вказану ним пошту, разом із паролем, який слід ввести у відповідне поле. У випадку, якщо введено неправильний пароль, спроба відновлення буде припинена.

Після цього ви отримаєте лист на електронну пошту із повідомленням про зміну пароля та унікальним ключем підтвердження. Цей ключ потрібно буде ввести у відповідне поле, рис. 4.20 та рис 4.21.

```
X-Sendmail-Commandline: sendmail.pl -t -i
To: bodya7919@gmail.com
Subject: Confirm that you want to change your email
X-PHP-Originating-Script: 0:recovery.php

Do you want to change your password?

Please, write this key in special field if you want change password key= d41d8cd98f00b204e9800998ecf8427e7914
```

Рисунок 4.20 – Пробний текст який, прийде на пошту

Рисунок 4.21 – Поля для вводу ключа та нового паролю

Як тільки ви все зробите правильно, введете ключ і новий пароль, вас відразу перенаправить на сторінку входу, де буде написано що ви успішно поміняли пароль, дякую, ви можете заново увійти в свій аккаунт.

4.4 Оптимізація та тестування

Оптимізація та тестування – це одне із самих головних дій, будь якого розробника ігри, перед випуском гри на світ, її дуже довго тестують і перевіряють на різні баги та проблеми, оптимізують під різні комп'ютери та графіку, намагаються щось нове додати, десь заощадити пам'ять.

Є закриті та відкриті бета тестування, коли гра готова, а також, можуть перевіряти на поведінку комп'ютера, відеокарт, на ті чи інші модельки персонажів, локацій тощо.

Етапи оптимізації гри:

1. оптимізація графіки та ресурсів – для забезпечення ефективного використання графіки та ресурсів, наприклад, можна спробувати зменшити розмір текстур та інших зображень;
2. асинхронні операції – це дії, виконані в неблокуючому режимі, що дозволяє основному потоку програми продовжити обробку;
3. профілювання коду – є такі програми, наприклад, в РНР можна використовувати Xdebug, він може, виявити місця в коді, та може запропонувати варіанти покращення коду для менших витрат пам'яті;
4. кешування – може бути використано для зберігання проміжних результатів обчислень або завдань, що не змінюються часто;
5. оптимізація алгоритмів – для покращення будь-якої гри, можна використовувати алгоритми, для використання більш ефективних вирішення підходів.

Етапи тестування гри:

1. функціональне тестування – перевірка на правильність виконання основних функцій гри;
2. тестування сумісності – тестування гри на різних конфігураціях, наприклад, на телефоні та на комп'ютері;
3. стрес-тестування – Робиться для перевірки на, як гра буде поводити себе, під час великого навантаження системи;
4. тестування продуктивності – вимірювання кількості кадрів в секунду, чим більше, тим краще;
5. локалізація та тестування мов – перевірка на сумісність з іншими мовами наприклад, англійська, українська, німецька;
6. тестування безпеки – робиться для того, щоб, дані користувача не можна було вкрати, для безпеки даних гри;
7. особливі випадки – це вже непередбачені випадки, можливо, якісь баги, на карті, або, текстури можуть провалитися;

8. автоматизоване тестування – це автоматизовані тести для автоматизації тестових сценаріїв та швидкого виявлення помилок.

В мене гра не такого масштабного рівня, тому, перевірів гру на основні функції, тобто, змійка, якщо вріжеться в себе або в стінку, гра закінчується, лічильник гри працює, їжу їсть.

Також, провів різні тести, наприклад, перевірів чи відправиться смс на пошту, при відновленні пароля, як дані можна з'ясувати, за результатами перевірки, дані повністю захищені

Багаторазово, треба повторювати оптимізацію та тестування під час розробки гри, оскільки це допомагає уникнути потенційних проблем та забезпечити високу якість продукту.

4.5 Висновки до розділу

У цьому розділі було створення бази даних MySQL в панелі управління phpMyAdmin, детально описано, кожен крок, чому вибрав назву колонки, тип, розмір і т.д. Саме підключення бази даних через мову програмування PHP через функцію `mysqli_connect()`, запис даних в базу даних users, створення форм входу в аккаунт, запис даних в БД, сторінка скидання паролю та його відновлення, поле для вводу ключа, були описані етапи оптимізації та тестування гри.

ВИСНОВКИ

Метою цього проекту розгляд можливостей мови програмування JS, HTML/CSS, SQL, забезпечити можливість взаємодії користувача сайту (гравця) з алгоритмом гри, а також збереження результату гри на сервері за допомогою мови програмування PHP.

В ході кваліфікаційної роботи було розглянуто засоби та технології створення ігор. В ході роботи було проведено аналіз існуючих мов програмування та було обрано наступні мови програмування JS, PHP, HTML, CSS, для створення бази даних – MySQL.

Після аналізу, вирішено створити гру для комп'ютера, з використанням сайту, перевага гри буде, в тому, що, буде база даних куди записуються дані про гравця та його обліковий запис, надалі, за допомогою цих даних, гравець може входити в свій акаунт і надалі вибирати ігри в які він гратиме.

В ході кваліфікаційної роботи було розроблено та описано концепт-документ гри, опис та розробка сюжету гри, мета та цільова аудиторія, опис ігроладу, аналіз жанру гри, терміни виконання та бюджет, майбутня перспектива. Проект розрахований для створення сайту з власної базою даних, полем реєстрації, щоб людина могла входити в акаунт і вибирати надалі у що буде грати. Гра призначена для дітей від 3+, а також і для дорослих, кожен вік може грати в неї.

Основною метою гравця є контроль невеликого об'єкта, який представляє собою змію, і розвивати його, їдаючи їжу, яка з'являється на ігровому полі. Одночасно гравець повинен уникати зіткнень зі своїм власним тілом та межами ігрового поля.

Цей проект був написаний в рамках кваліфікаційної роботи, гра була розроблена для опрацювання основних шагів створення ігор, тут було опрацьовано основні моменти і концепції як створювати і на які кроки потрібно звертати увагу. Надалі, завдяки цим знанням, можна буде створювати ігри на різні моделі, наприклад, на комп'ютери та на телефони.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Офіційний сайт World Wide Web Consortium [Електронний ресурс]
URL: <https://www.w3.org> (дата звернення 25.11.2023);
2. Офіційний сайт PHP. Hypertext Preprocessor [Електронний ресурс];
URL: <https://www.php.net> (дата звернення 25.11.2023);
3. Офіційний сайт Сучасний підручник JavaScript [Електронний ресурс]
URL: <https://learn.javascript> (дата звернення 26.11.2023);
4. Сайт 15 кращих HTML5 фреймворків [Електронний ресурс]
URL: <https://ruseller.com/lessons.php> (дата звернення 26.11.2023);
5. Сайт Топ-9 JS-движків та бібліотек для ігор у 2020 році [Електронний ресурс]
URL: <https://vc.ru/dev/131649-top-9-js-dvizhkov-i-bibliotek-dlya-igr-v-2020-godu> (дата звернення 26.11.2023);
6. Офіційний сайт Valve developer community [Електронний ресурс]
URL: <https://developer.valvesoftware.com> (дата звернення 30.11.2023);
7. Офіційний сайт Resources for Developers, by Developers [Електронний ресурс]
URL: <https://developer.mozilla.org> (дата звернення 26.11.2023);
8. Офіційний сайт Анатомія відеоігри [Електронний ресурс]
URL: <https://developer.mozilla.org/docs/Games/Anatomy> (дата звернення 24.11.2023);
9. Офіційний сайт SQL підручник [Електронний ресурс]
URL: <https://html5css.ru/sql/default.php> (дата звернення 25.11.2023);
10. Developing and Applying Optoelectronics in Machine Vision/ O. Sergiyenko, J.C. Rodriguez-Quiñonez, IGI Global, 2016; 341p.
11. Офіційний сайт HTML плюси та мінуси [Електронний ресурс]
URL: <https://kztarif/kompjutyery/html-pljusy-i-minusy> (дата звернення 25.11.2023);
12. Карпухін О.В. - Комплекс навчально-методичного забезпечення навчальної дисципліни «Internet- технології» 270 с;

13. https://www.youtube.com/watch?v=eQiAt-Mf_HU&list=PLm-m_n0n5oBhhUAPMjneL3xdOU2U5dNZz - IT Essentials v7 (ITE) Cluster;
14. С.М. Порошин, В.М. Карташов, В.В. Усик, Р.І. Цехмістро, І.С.Беліков. Технології створення складових мультимедійного контенту. Анімація та web-анімація. Навчальний посібник –Харків, НТУ.ХПІ, 2022.-314с.
15. М.А. Омаров Основи технологій сучасної web-анімації. Навчальний посібник /М.А. Омаров, В.М. Карташов, Р.І. Цехмістро, В.В. Усик – Харків ХНУРЕ- 2022р.- 214с
16. Офіційний сайт Питання з позначкою javascript [Електронний ресурс] URL: <https://stackoverflow.com/questions/tagged/javascript> (дата звернення 18.11.2023);
17. Офіційний сайт Создание сервера для онлайн ММО игр на PHP ч. 12 — Очереди и параллельное программирование на CPU [Електронний ресурс] URL: <https://habr.com/ru/articles/725490/> (дата звернення 19.11.2023);
18. V. M. Kartashov, V. N. Oleynikov, S. A. Sheyko, I. V. Koryttsev, S. I. Babkin, O. V. Zubkov, "Peculiarities of small unmanned aerial vehicles detection and recognition," *Telecommunications and Radio Engineering*, 2019, V. 78, Iss. 9, pp. 771–781. DOI: 10.1615/TelecomRadEng.v78.i9.30.
19. V. N. Oleynikov, O. V. Zubkov, V. M. Kartashov, I. V. Korytsev, S. I. Babkin, S.A. Sheiko, "Investigation of detection and recognition efficiency of small unmanned aerial vehicles on their acoustic radiation," *Telecommunications and Radio Engineering*, 2019, V. 78, Iss. 9, pp. 759–770. DOI: 10.1615/TelecomRadEng.v78.i9.20.
20. Kartashov, V.M., Sidorov, G.I., Sheiko, S.A., Kolendovska, M.M., Sergienko O.Yu. Principles of Construction and Assessment of technical Characteristics of multi-Frequency atmospheric Sodar in the Humidity Measurement Mode / *Telecommunications and Radio Engineering*.- New York. - 2020.- Vol. 79, №4.- P.323-333. (стаття). DOI: 10.1615/TelecomRadEng.v79.i4.50.

21. Kartashov, V.M., Oleynikov V.N, Zubkov, O.V., Korytsev I.V., Babkin, S. I., Sheiko, S.A., Kolendovskaya, M.M. Spatial-temporal Processing of acoustic Signals of Unmanned Aerial Vehicles; Telecommunications and Radio Engineering, 2020. Vol. 79, Iss, 9, pp.769-780.
22. V.M. Semenets, V.M. Kartashov, V.I. Leonidov. Features of Acoustic Noise of Small Unmanned Aerial Vehicles // Telecommunications and Radio Engineering.- New York. - 2020.- Vol. 79, №11.- P. 985-995. DOI: 10.1615/TelecomRadEng.v79.i11.80 (стаття).
23. Oleynikov V.N., Kartashov, V.M., Babkin, S. I., Zubkov, O.V., Korytsev I.V., Sheiko, S.A., Seleznov I.S. Structure and Parameter Unmanned Aerial Vehicles Sound Fields/ Telecommunications and Radio Engineering.- New York. - 2020.- Vol. 79, №17.- P.1539-1550. DOI: 10.1615/TelecomRadEng.v79.i17.50 (стаття).
24. В.А. Тихонов, В.М. Карташов, В.М. Олейников, В.І. Леонідів, Л.П. Тимошенко, І.С. Селезнев, Н.В. Рибніков. Виявлення-розпізнавання безпілотних літальних апаратів з використанням складової моделі авторегресії їхнього акустичного випромінювання// Вісник НТУУ «КПІ». Радіотехніка. Радіоапаратобудування. – 2020. – Вип. №81. - С.38-46. (Web of Science) Карташов В.М., Коритцев І.В., Олійников В.М., Зубков О.В., Шейко С.А., Бабкін С.І., Левський Н.А., Селезньов І.С. Алгоритми пеленгації безпілотних літальних апаратів з їхнього акустичного випромінювання// Радіотехніка. (Харків). - 2019. - Вип. 196. - С. 22-31. Карташов В.М., Харченко О.І., Чумаков В.І. Використання ефекту стохастичного резонансу аналізу спектрів акустичного випромінювання малих безпілотних літальних апаратів // Радіотехніка. (Харьков). — 2019. — Вып. 197. — С. 100-106.
25. Карташов В.М., Сидоров Г.І., Толстих Є.Г., Шаповалов С.В. Акустичний вимірювач швидкості вітру в атмосферному прикордонному шарі// Радіотехніка. (Харьков). — 2019. — Вып. 199. — С. 54-58.

26. A Comparative Example Between The Use Of Pca And Mds For Image Classification / Hernandez, W., Mendez, A., Flor-Unda, O., Camejo, I.M., Kolendovska, M. // IEEE International Symposium on Industrial Electronics, 29th IEEE International Symposium on Industrial Electronics, ISIE 2020; Delft; Netherlands; 17 June 2020 до 19 June 2020; Volume 2020-June, June 2020, № 9152565, Pages 1353-1358
27. Algorithm For Generating Refined Frequency Estimates In Atmospheric Radio Sounding Systems / Kartashov V., Hernandez W., Hernandez-Balbuena D., M. Kolendovska, Konovalenko O., Melnyk V. // IEEE International Symposium on Industrial Electronics, 29th IEEE International Symposium on Industrial Electronics, ISIE 2020; Delft; Netherlands; 17 June 2020 до 19 June 2020; Volume 2020-June, June 2020, № 9152562, Pages 79-82
28. Application of Fast Frequency Shift Measurement Method for INS in Navigation of Drones / D. Avalos-Gonzalez, D.H. Balbuena, V. Tyrsa, V.M. Kartashov, M. Kolendovska, S. Sheiko, O. Sergiyenko, V. Melnyk, F.N. Murrieta-Rico // IECON 2018 – 44th Annual Conference of the IEEE Industrial Electronics Society. – P. 3159–3164.
29. Avalos-Gonzalez, D., Sergiyenko, O., Hernandez-Balbuena, D., Tyrsa, V., Kartashov V.M., V., Rivas-Lopes, M., Murrieta-Rico, F.N. Constraints definition and application optimization based on geometric analysis of the frequency measurement method by pulse coincidence // Measurement: Journal of the International Measurement Confederation (USA). 2018, V.126. P. 184-193.
30. Book “Control and Signal Processing Applications for Mobile and Aerial Robotic Systems”, Hardback - Advances in Computational Intelligence and Robotics English. Edited by Oleg Sergiyenko, Moises Rivas-Lopez, Wendy Flores-Fuentes, Julio Cesar Rodríguez-Quíñonez, Lars Lindner. Editorial IGI Global, Hershey, United States, January 2020, 340 páginas. ISBN10 152259924X, ISBN13 9781522599241

31. Cesar Sepulveda-Valdez ; Oleg Sergiyenko ; Vera Tyrsa ; Wendy Flores-Fuentes ; Julio César Rodríguez-Quíñonez ; Fabian Natanael Murriente-Rico ; Jesús Elías Miranda-Vega ; Paolo Mercorelli ; Marina Kolendovska. "Geometric analysis of a laser scanner functioning based on dynamic triangulation," 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 1398-1403, doi: 10.1109/ISIE45063.2020.9152268.
<https://ieeexplore.ieee.org/abstract/document/9152268>
32. Cuauhtémoc Mariscal-García; Wendy Flores-Fuentes; Daniel Hernández-Balbuena; Julio C. Rodríguez-Quíñonez ; Oleg Sergiyenko. "Classification of Vehicle Images through Deep Neural Networks for Camera View Position Selection," 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 1376-1380, doi: 10.1109/ISIE45063.2020.9152440.
<https://ieeexplore.ieee.org/abstract/document/9152440>
33. Developing and Applying Optoelectronics in Machine Vision/ O. Sergiyenko, J.C. Rodriguez-Quíñonez, IGI Global, 2016; 341p.
34. Experimental estimation of direction finding to unmanned air vehicles algorithms efficiency by their acoustic emission, /Oleynikov, V., Zubkov, O., Kartashov, V., ...Sheiko, S., Babkin, S.//2019 IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology, PIC S and T 2019 - Proceedings, 2019, стр. 175-178, 9061337
35. Features of acoustic noise of small unmanned aerial vehicles / Semenets, V.V., Kartashov, V.M., Leonidov, V.I. //Telecommunications and Radio Engineering (English translation of *Elektrosvyaz* and *Radiotekhnika*), 2020, 79(11), стр. 985-995
36. Geometric Analysis Of A Laser Scanner Functioning Based On Dynamic Triangulation /Sepulveda-Valdez, C., Sergiyenko, O., Tyrsa, V, Mercorelli, P., Kolendovska, M.// IEEE International Symposium on Industrial

- Electronics, 29th IEEE International Symposium on Industrial Electronics, ISIE 2020; Delft; Netherlands; 17 June 2020 до 19 June 2020; Volume 2020-June, June 2020, № 9152268, Pages 1398-1403
37. <https://ieeexplore.ieee.org/abstract/document/9152255>
38. <https://ieeexplore.ieee.org/document/9161870>
39. I. Y. A. Corpus, L. Lindner, O. Sergiyenko. "Transimpedance Amplifier for Laser Scanning System Range Extension," 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 1421-1426, doi: 10.1109/ISIE45063.2020.9152487. <https://ieeexplore.ieee.org/abstract/document/9152487>
40. Ivanov, M., Sergiyenko, O., Mercorelli, P., Hernandez, W.c, Rodriguez Quinonez, J.C.d, Katashov V., Kolendovska, M., Iryna, T. Effective informational entropy reduction in multi-robot systems based on real-time TVS. IEEE International Symposium on Industrial Electronics, 2019-June, 8781209, c. 1162-1167.
41. Jonathan J. Sanchez-Castro ; Julio C. Rodríguez-Quiñonez ; Luis R. Ramírez-Hernández ; Guillermo Galaviz ; Daniel Hernández-Balbuena ; Gabriel Trujillo-Hernández ; Wendy Flores-Fuentes ; Paolo Mercorelli ; Wilmar Hernández-Perdomo ; Oleg Sergiyenko ; Félix Fernando González-Navarro. "A Lean Convolutional Neural Network for Vehicle Classification," 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 1365-1369, doi: 10.1109/ISIE45063.2020.9152274. <https://ieeexplore.ieee.org/abstract/document/9152274>
42. Lindner, L., Sergiyenko, O., Rivas-López, M., (...), Gurko, A., Kartashov, V.M. Machine vision system for UAV navigation; IEEE, 2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles and International Transportation Electrification Conference, ESARS-ITEC, 2016; pp.1–6. DOI: 10.1109/ESARS-ITEC.2016.7841356.

43. M. Ivanov, O. Sergiyenko, V. Tyrsa, P. Mercorelli, V. Kartashov, W. Hernandez, S. Sheiko, M. Kolendovska. Individual scans fusion in virtual knowledge base for navigation of mobile robotic group with 3D TVS // Proceedings of 44th Annual Conference of IEEE Industrial Electronics Society (IECON).. -2018. – Washington DC, USA. -S. 3187-3192. . ISBN 978-1-5090-6683-4/18/.
44. Murrieta-Rico, F.N., Petranovskii, V., Galvan, D.H., Sergiyenko, O., Yocupicio-Gaxiola, R.I., De Dios Sanchez-Lopez, J. Phase effect in frequency measurements of a quartz crystal using the pulse coincidence principle. 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 185-190, 9152255, DOI: 10.1109/ISIE45063.2020.9152255
45. Oleksandr Sotnikov, Vladimir Kartashov, Oleksandr Tymochko, Oleg Sergiyenko, Vera Tyrsa, Paolo Mercorelli, Wendy Flores-Fuentes. Methods for Ensuring the Accuracy of Radiometric and Optoelectronic Navigation Systems of Flying Robots in a Developed Infrastructure. Chapter 16// Machine Vision and Navigation; Springer, Cham. pp.537–578. Editors: Sergiyenko, Oleg, Flores-Fuentes, Wendy, Mercorelli, Paolo. DOI: 10.1007/978-3-030-22587-2_16.
46. Optical detection of unmanned air vehicles on a video stream in a real-time/Kartashov, V., Oleynikov, V., Zubkov, O., Sheiko, S.// 2019 International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019 - Proceedings, 2019, 9165362/
47. Principles Of Construction And Assessment Of Technical Characteristics Of Multi-Frequency Atmospheric Sodar In The Humidity Measurement Mode / Kartashov, V.M., Sidorov, G.I., Sheiko, S.A., Kolendovskaya, M.M., Sergienko, O.Yu. // Telecommunications And Radio Engineering (English Translation Of Elektrosvyaz And Radiotekhnika), 2020, ISSN

- Print: 0040-2508, ISSN Online: 1943-6009,
DOI: 10.1615/TelecomRadEng.v79.i4.50, p. 323-333/
48. Research Of The Uncertainty Of Measurement Frequencies And Definitions Of The Frequency Signal In The Waveguide With Respect To Power / Semenets, V.Zakharov, I. Serhienko, M., Kartashov, V.M., Kolendovska, M., Hernandez, W., Hipolito, J.I.N., Tyrsa, V.// 45th Annual Conference of the IEEE Industrial Electronics Society, IECON 2019; Lisbon Congress CenterLisbon; Portugal; 14 October 2019 до 17 October 2019; CFP19IEC-ART; Код 155980, Volume 2019-October, October 2019, № 8927203, Pages 4674-4679
49. Spatial-Temporal Processing Of Acoustic Signals Of Unmanned Aerial Vehicles /Kartashov V.M., Oleinikov V.N., Zubkov O.V., Sheiko S.A., Kolendovska M.M.// Telecommunications And Radio Engineering (English Translation Of Elektrosvyaz And Radiotekhnika), 2020, ISSN Print: 0040-2508, ISSN Online: 1943-6009, DOI: 10.1615/Telecomradeng.v79.i9.40, p. 769-780
50. Stereoscopic Vision Systems In Machine Vision, Models, And Applications (Book Chapter)/ Ramírez-Hernández, L.R., Rodríguez-Quiñonez, J.C., Castro-Toscano, M.J., Kolendovska, M., Murrieta-Rico, F.N.// Machine Vision And Navigation, 2019 Machine Vision and Navigation 30 September 2019, Pages 241-265
51. Strelkova T., Kartashov V., Lytyuga A., Strelkov A. Theoretical Methods of Images Processing in Optoelectronic Systems. Chapter 16. // Biometrics: Concepts, Methodologies, Tools, and Applications; Oleg Sergiyenko and Julio C. Rodriguez-Quiñonez. (341p.), IGI Global, 2017; pp. 361-381. DOI: 10.4018/978-1-5225-0983-7.ch016.
52. Strelkova T., Kartashov V., Lytyuga A., Strelkov A. Theoretical Methods of Images Processing in Optoelectronic Systems. Chapter 6// Developing and Applying Optoelectronics in Machine Vision; Oleg Sergiyenko and Julio C. Rodriguez-Quiñonez. (341p.) – USA, Herhey, IGI Global, 2016; pp.180-205.

53. Sytnik O., Kartashov V. Methods and Algorithms for Technical Vision in Radar Introspection. Chapter 13// Optoelectronics in Machine Vision-Based Theories and Applications. IGI Global, 2019; pp. 373-391.
54. The Use of Factorization and Multimode Parametric Spectra in Estimating Frequency and Spectral Parameters of Signal/Semenets, V., Kartashov, V., Sergiyenko, O., ...Rodriguez-Quinonez, J.C., Flores-Fuentes, W.//IEEE International Symposium on Industrial Electronics, 2020, 2020-June, p. 215-219
55. Unda, O.F., Hernandez, W., Vargas, O., Mendez, A., Sergiyenko, O., Tyrsa, V. Construction of a robotic platform of differential type for first-year students of electronic engineering, 2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion, SPEEDAM 2020, 24-26 de junio de 2020, Sorrento, Italia, pp. 538-543, 9161870, DOI: 10.1109/SPEEDAM48782.2020.9161870
56. Use of Acoustic Signature for Detection, Recognition and Direction Finding of Small Unmanned Aerial Vehicles/Kartashov, V., Oleynikov, V., Koryttsev, I., ...Babkin, S., Selieznov, I.//Proceedings - 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2020, 2020, p. 377-380/
57. V. Semenets; Vladimir Kartashov ; Oleg Sergiyenko; Vyacheslav Tikhonov ; Paolo Mercorelli ; Sergiy Sheiko ; Nataliya Chmelarova. "The Use of Factorization and Multimode Parametric Spectra in Estimating Frequency and Spectral Parameters of Signal," 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 215-219, doi: 10.1109/ISIE45063.2020.9152238.
<https://ieeexplore.ieee.org/abstract/document/9152238>
58. Wilmar Hernandez ; Alfredo Mendez ; Omar Flor-Unda ; Vicente Gonzalez-Posada ; Jose Luis Jimenez ; Oleg Sergiyenko ; Julio C. Rodriguez-Quinonez ; Mykhailo Ivanov ; Ivan Menes Camejo ; Marina Kolendovska. "A comparative example between the use of PCA and MDS for image

classification," 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 1353-1358, doi: 10.1109/ISIE45063.2020.9152565.

<https://ieeexplore.ieee.org/abstract/document/9152565>