

Галицький коледж імені В'ячеслава Чорновола  
відділення комп'ютерних та видавничих технологій  
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач відділенням  
комп'ютерних та видавничих  
технологій  
Чубей О.О. / \_\_\_\_\_ /  
підпис  
« \_\_\_\_ » \_\_\_\_\_ 2021 р.

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту  
освітньо-кваліфікаційного рівня «молодший спеціаліст»  
зі спеціальності 122 «Комп'ютерні науки»  
на тему: «Ігровий мобільний додаток в жанрі “Аркада”»

Студент групи К-47	Головатий А.П.	_____
		(підпис)

Керівник проєкту	Гавришків Н.Г.	_____
		(підпис)

Консультанти:

з техніко-економічного обґрунтування	Меленчук Л.І.	_____
		(підпис)

нормоконтролер	Кульчинська Н.З.	_____
		(підпис)

Тернопіль – 2021

Галицький коледж імені В'ячеслава Чорновола  
відділення комп'ютерних та видавничих технологій  
циклова комісія інформатики та комп'ютерних дисциплін

**ЗАТВЕРДЖУЮ**

Завідувач відділенням  
комп'ютерних та видавничих  
технологій

Чубей О.О. / \_\_\_\_\_ /

підпис

« \_\_\_\_ » \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ**

на дипломне проєктування  
на здобуття освітньо-кваліфікаційного рівня «молодший спеціаліст»  
студенту Головатому Андрію Петровичу  
\_\_\_\_\_  
(прізвище, ім'я та по-батькові студента)

1. Тема проєкту \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

затверджено наказом по коледжу від “ \_\_\_\_ ” \_\_\_\_\_ 202\_ р., № \_\_\_\_

2. Термін здачі студентом завершеного проєкту “ \_\_\_\_ ” \_\_\_\_\_ 202\_ р.

3. Вихідні дані до проєкту \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

4. Перелік питань, які повинні бути розроблені в проєкті:

а) основна частина \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

б) техніко-економічне обґрунтування \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

5. Перелік графічного матеріалу \_\_\_\_\_  
\_\_\_\_\_

6. Консультанти проекту: \_\_\_\_\_

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко-економічного обґрунтування	_____		
	(вчена ступінь, звання П.І.Б. консультанта)		

### КАЛЕНДАРНИЙ ПЛАН дипломного проектування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до дипломного проектування	16.11.20	30.11.20
2.	Огляд типових рішень та написання відповідного розділу ПЗ	01.12.20	26.01.21
3.	Дослідження технологій реалізації та написання відповідного розділу ПЗ	27.01.21	15.02.21
4.	Розробка функціональних вимог до проекту та робота над структурою програмного продукту. Написання відповідного розділу ПЗ	15.02.21	02.03.21
5.	Встановлення та налаштування середовища реалізації та написання відповідного розділу ПЗ	02.03.21	16.03.21
6.	Проектування програмного засобу (функціоналу, інтерфейсу, бази даних продукту) та написання відповідного розділу ПЗ	16.03.21	16.04.21
7.	Реалізація та налаштування програмного засобу та написання відповідного розділу ПЗ	17.04.21	03.05.21
8.	Доопрацювання модулів	03.05.21	17.05.21
9.	Опрацювання економічного розділу дипломного проекту та оформлення спеціального розділу	17.05.21	18.06.21
10.	Тестування та налагодження програмного продукту та написання відповідного розділу ПЗ	18.05.21	04.06.21
11.	Робота над оформленням пояснювальної записки	04.06.21	11.06.21
12.	Попередній захист дипломного проекту, доопрацювання	11.06.21	
13.	Підготовка до захисту дипломного проекту	18.06.21	23.06.21
14.	Захист дипломного проекту	23.06.21	23.06.21

7. Дата видачі “\_\_\_” \_\_\_\_\_ 2020р. Керівник \_\_\_\_\_ /

Завдання прийняв до виконання \_\_\_\_\_ /

## Реферат

Ігровий мобільний додаток в жанрі “Аркада”. Дипломний проєкт. Головатий Андрій Петрович. Галицький коледж імені В’ячеслава Чорновола, відділення комп’ютерних та видавничих технологій. Спеціальність 122 «Комп’ютерні науки». ГК, 2021. Сторінок – 49, рисунків – 29, додатків – 1.

У першому розділі дипломної роботи розглянуто аналіз предметної області та розглянено декілька існуючих рішень.

У другому розділі здійснено формалізацію вимог до системи, вимог до предметного продукту, Виконано проєктування системи та її складових, а також описано алгоритм.

Третій розділ розкриває технології засобів реалізації, містить опис розробки і тестування програми.

Метою проєкту є реалізація відеогри, а саме ключових механік таких як пересування персонажем, знищення об’єктів.

Результатом роботи над проєктом є повністю оптимізована та робоча відеогра.

ВІДЕОІГРИ, МУЛЬТИМЕДІА, МОБІЛЬНИЙ ДОДАТОК, АЛГОРИТМ, ГЕЙМПЛЕЙ, СИСТЕМА, ГРАВЕЦЬ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, КЛЮЧ, АРКАДА.

## Abstract

Arcade game mobile application. Diploma project. Golovaty Andrey Petrovich. Vyacheslav Chornovil Galician College, Department of Computer and Publishing Technologies. Specialty 122 "Computer Science". GC, 2021. Pages – 49, figures – 29, appendixes – 1.

In the first section of the thesis the analysis of the subject area is considered and several existing decisions are considered.

In the second section the formalization of system requirements, requirements to the subject product is carried out, the system and its components are designed, and the algorithm is described.

The third section reveals the technology of implementation tools, contains a description of the development and testing of the program.

The aim of the project is to implement a video game, namely key mechanics such as character movement, destruction of objects.

The result of working on the project is a fully optimized and working video game.

VIDEO GAMES, MULTIMEDIA, MOBILE APP, ALGORITHM, GAME, SYSTEM, PLAYER, SOFTWARE, KEY, ARCADE.

## ЗМІСТ

Вступ.....	7
1 Аналіз існуючих рішень та постановка завдання .....	8
1.1 Актуальність теми.....	8
1.2 Аналіз наявних рішень .....	12
1.3 Постановка задачі .....	16
2 Проєктування системи.....	17
2.1 Формалізація вимог до системи .....	17
2.2 Проєктування структури системи .....	17
2.3 Просування алгоритму роботи системи .....	18
3 Реалізація та тестування системи .....	21
3.1 Аналіз інструментальних засобів для реалізації гри.....	21
3.2 Вибір засобів реалізації .....	21
3.3 Вибір допоміжного програмного забезпечення.....	25
3.4 Створення гри.....	25
4 Технічно економічне обґрунтування .....	35
4.1 Аналіз ринку .....	35
4.2 Розрахування витрат на розробку проєкту.....	36
4.3 Обґрунтування необхідності розробки.....	37
Висновки .....	38
Перелік джерел посилання .....	39
Додатки.....	40

					<i>ДП. КН 21.434.13.000 ПЗ</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Ігровий мобільний додаток в жанрі “Аркада”	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Розроб.</i>		Головатий А.П						
<i>Перевір.</i>		Гавришків Н.Г					5	49
<i>Реценз..</i>		Посвятовська О.Б				ГК. КВТ. К-47		
<i>Н.контр.</i>		Кульчинська Н.З						
<i>Зав. відділ.</i>		Чубей О.О						

## СКОРОЧЕННЯ І УМОВНІ ПОЗНАКИ

MP3 – MPEG-1 Audio Layer 3

AMR – Adaptive Multi-Rate Narrow Band (AMR-NB),

MIDI – Musical Instrument Digital Interface,

WAV – waveform audio format,

JPEG – Joint Photographic Experts Group,

BMP – bevice independent bitmap,

WYSIWYG – What You See Is What You Get.

					<i>ДП. КН 21.434.13.000 ПЗ</i>	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Відеогра – це електронний додаток, у якому користувач, а саме гравець використовує інтерфейс програми. Електронний пристрій називається платформером. До них належать персональні комп'ютери, ігрові платформи та мобільні пристрої. Пристрій завдяки якому відбувається керування на екрані, зветься ігровим контроллером.

Одними з перших аркадних платформерів були: “Frogs” (1978) були стрибки, але не було платформ та “Space Panic” (1980) де були платформи, але не було стрибків. Першим справжнім платформером стала гра “Pitfall”, та все ж слава прийшла після виходу “Super Mario Bros” [1].

За останній період часу, відеоігри набули великої популярності, майже у кожного власника мобільного пристрою, є одна або більше відеоігор. Це допомагає людям швидше згаяти час, наприклад при очікуванні чогось, тощо. Також відеоігри приносять користувачам задоволення, завдяки чому, можна весело проводити час і на одинці.

Метою розробки є використання теоретичних відомостей про сучасні засоби, методи для побудови відеоігри, у якому буде реалізована система керування персонажем, можливість взаємодіяти з об'єктами та знищення об'єктів.

					ДП. КН 21.434.13.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ПОСТАНОВКА ЗАВДАННЯ

## 1.1 Актуальність теми

Android – це вільна ОС. Використовується у таких пристроях як мобільні телефони, планшетні комп'ютери, розумні годинники, смартбуків та у телевізорах. Використовує ядро “Linux” [2]. Перша версія була випущена у вересню 2008 р. Уже в 2008 році, ОС получила біля 40 оновлень. На рисунку 1.1 зображена послідовність оновлення версій.

Обновления версий Android

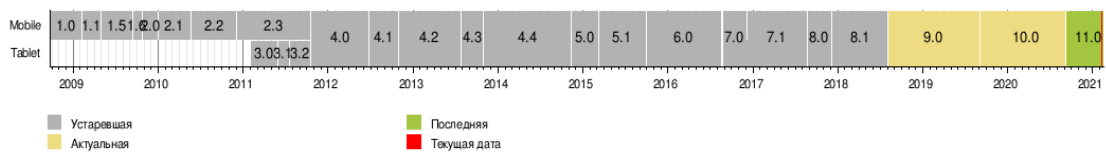


Рисунок 1.1 – Графік оновлень ОС

Модифікувати ОС дозволяє відкритий код, створювати нові додатки, модифікувати операційну систему. Android без клопотання встановлюється на абсолютно довільний мобільний пристрій (смартфон). Рисунок 1.2 – логотип Android.



Рисунок 1.2 – Логотип Android

Однією з перших версій цієї ОС була 1.0 - Android 1.0 Apple Pie, випущена 23 вересня 2008 року це була перша стабільна версія. Рисунок 1.3.

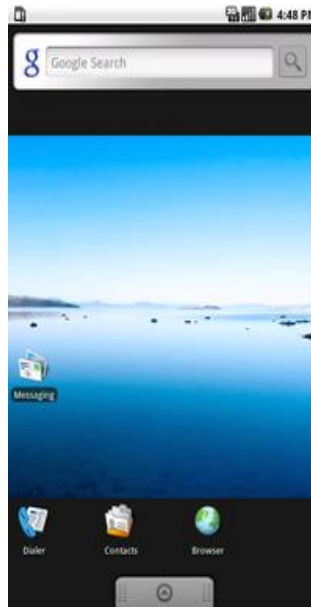


Рисунок 1.3 – Інтерфейс Android 1.0

Через певний проміжок часу, а саме 10 травня 2011 року був анонсований Android 4.0. У даній версії була покращена багатозадачність та була замінена панель з сповіщеннями, також була реалізована можливість створювати папки на робочому столі, що зображено на рисунку 1.4.



Рисунок 1.4 – Інтерфейс Android 4.0

					ДП. КН 21.434.13.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Одна з найпопулярніший версій зображена на рисунку 1.5 - Android 7.0. Дана версія получила чудові нові функції, одною з таких функцій є можливість розділити екран навпіл, іншими словами виконувати декілька задач одночасно, також був створений новий дизайн для папок.

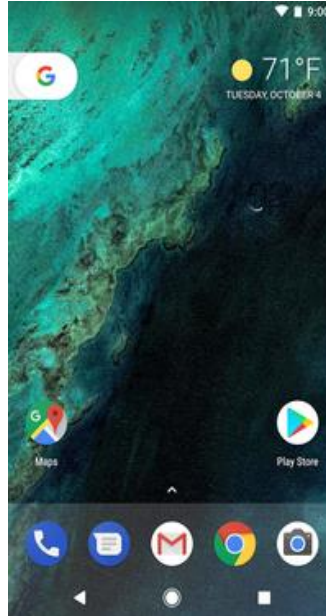


Рисунок 1.5 – Інтерфейс Android 7.0

8 вересня 2020 виходить нова версія - Android 11, рисунок 1.6. Головним нововведенням була підтримка 5G мережі, меню сповіщень було розділено, створенням нової панелі сповіщень у меню “ромови” [3].

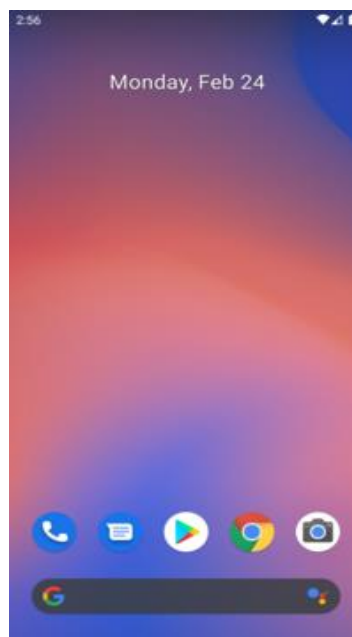


Рисунок 1.6 – Інтерфейс Android 11.0

					<i>ДП. КН 21.434.13.000 ПЗ</i>	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Компанія Google видавала назви версіям з принципу кулінарії рисунок 1.7.



Рисунок 1.7 – Принцип кулінарії

Можливості операційної системи Android:

- Обмін повідомленнями.
- Підтримка декількох мов.
- Web браузер.
- Підтримка Java.
- Багатозадачність.
- Можливість передавати дані на різноманітні накопичувачі.
- Підтримка медіа файлів (MP3, AMR, MIDI, WAV, JPEG, BMP, WebP H.263 та інші ).

Екран містить в собі кілька сторінок з різною інформацією. Рядок стану знаходиться на самому верху, він інформує про пристрій та різноманітні підключення. Цей рядок можна перетягти униз, для отримання інформації про можливі підключення, та швидкого налаштування пристрою, також він відображає екран повідомлень, іншу корисну інформацію, тощо.

Популярність мобільних ігор зростає. Ринок мобільних ігор за останні роки стрімко розвивається. Експерти стверджували, що на 2021 рік мобільні ігри становитимуть 60% прибутку ринку. На сьогоднішній день не мала кількість ІТ-компаній зосереджує зусилля створенні мобільних ігор. Попит на цей продукт зараз на високому рівні і не мала кількість розробників перейшла на створення ігрових додатків. Через стрімке зростання популярності такі компанії як Facebook, Instagram і Twitter, розпочали розробку інноваційних мобільних ігор, вважаючи їх рушійною силою в залучені користувачів на свої майданчики.

Функціональність сучасних мобільних телефонів виросла настільки, що вони стали своєрідними міні-комп'ютерами.

Хороші екрани з хорошою роздільною здатністю, хорошим рівнем передавання кольору.

- Високошвидкісний інтернет.
- Web браузер.
- Сенсорний дисплей.
- Відкритий ринок.
- Доступність, абсолютно любий розробник з легкістю може опублікувати свій додаток.

Через різкий ріст популярності конкуренція на ігровому ринку винятково висока.

## 1.2 Аналіз наявних рішень

Аркада - це відеоігри з примітивним, водночас цікавим ігровим процесом. Є не мало видів аркад.

- Аркадні гонки.
- Класичні аркади.
- Бійки, боротьба.
- Платформери.
- Віртуальний тир.

					ДП. КН 21.434.13.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Платформер – це один із найбільш популярних видів аркад. Завдання гравця – подолання перешкод за допомогою стрибків, тощо. Гравцеві доводиться стрибати по абстрактно розставленими в повітрі об’єктами “платформам”, звідки була взята і назва.

Дивовижний платформер “Vector”. Це дуже захоплюча гра в аркадному стилі, гравець зображений у вигляді вільного бігуна, який не піддається системі [4].

Перевагами гри є:

- Доступна безкоштовна версія.
- Реалістичний паркур завдяки дивовижній анімації.
- 40 цікавих рівнів.
- Проста в освоєнні.

Недоліками гри є.

- Платний контент.
- Відсутня Українська локалізація.
- Багато рутини.

Кадри з відеогри зображенні на рисунку 1.8.



Рисунок 1.8 – Кадри з гри “Vector”

Sonic the Hedgehog – це неперевершена гра, гравець в ролі їжака Соніка. Він повинен проходити лабіринти, знищувати ворогів – роботів-бандитів Доктора Еггмана. Кадри з гри зображені на рисунку 1.9 [5].

Перевагами гри є:

- Доступна безкоштовна версія.
- Безліч цікавих рівнів.
- Цікавий сюжет.
- Проста в освоєнні.

Недоліками гри є:

- Відсутня Українська локалізація.



Рисунок 1.10 – Кадри з гри “Sonic the Hedgehog”

На рисунку 1.11 зображена одна з найвеселіших аркад - Fast like a Fox. Метою гри є пошук скарбу племені, а саме: монет, смарагдів та алмазів. Гравець стрибатиме поміж світами щоб знайти скарб і повернути його законним власникам [6].

Перевагами гри є:

- Доступна безкоштовна версія;
- Цікавий сюжет;
- Проста в освоєнні;

Недоліком гри є:

- Відсутня Українська локалізація.



Рисунок 1.11 – “Кадри з гри Fast like a Fox”

Super Bob's World 2020. Мета гри врятувати принцесу з джунглів перестрибуючи перешкоди та злих монстрів [7].

Кадри з гри зображені на рисунку 1.12.

Перевагами гри є:

- Доступна безкоштовна версія.
- Цікавий сюжет.
- Проста в освоєнні.

Недоліками гри є:

- Платний контент.
- Відсутня Українська локалізація.
- Багато рутини.



Рисунок 1.12 – “Super Bob`s World 2020”

### 1.3 Постановка задачі

Платформер, гравець керує персонажем, що буде рухатися рухомими або не рухомими платформами різноманітної висоти. Персонаж буде рухатися лише праворуч та вгору уминаючи різноманітні перешкоди. У гравця буде 3 життя. Метою гри буде отримання якомога більшої кількості очків та монет.

Результатом буде мобільна гра, де ви керуєте персонажем, який ухиляється від різноманітної небезпеки. Перешкоди наближаються з різноманітною швидкістю, гравець може керувати лише висотою персонажа ( торканням екрана на мобільному пристрої ). Рівень складається з нескінченного потоку, генерації перешкод.

Змн.	Арк.	№ докум.	Підпис	Дата

ДП. КН 21.434.13.000 ПЗ

Арк.

16

## 2 ПРОЄКТУВАННЯ СИСТЕМИ

### 2.1 Формалізація вимог до системи

#### Вимоги до системи

Мобільний додаток орієнтована на дітей від 6 років, буде зрозумілим для користування. Основною вимогою є простота в освоєнні. Простий та зрозумілий геймплей. Фонове зображення та персонажі, не повинні бути “різкими”. Зображення будуть у науково-фантастичному стилі.

Адаптація додатку до різноманітних смартфонів буде здійснюватись завдяки функції “ZoomEven”. Вона дозволить насолодитися грою на будь якому смартфоні.

### 2.2 Проєктування структури системи

У додатку будуть реалізовані такі можливості як: керування персонажем, створення рівнів, зміна текстур, системи керування, інтерфейс, збереження результатів. Декомпозиція проєкту

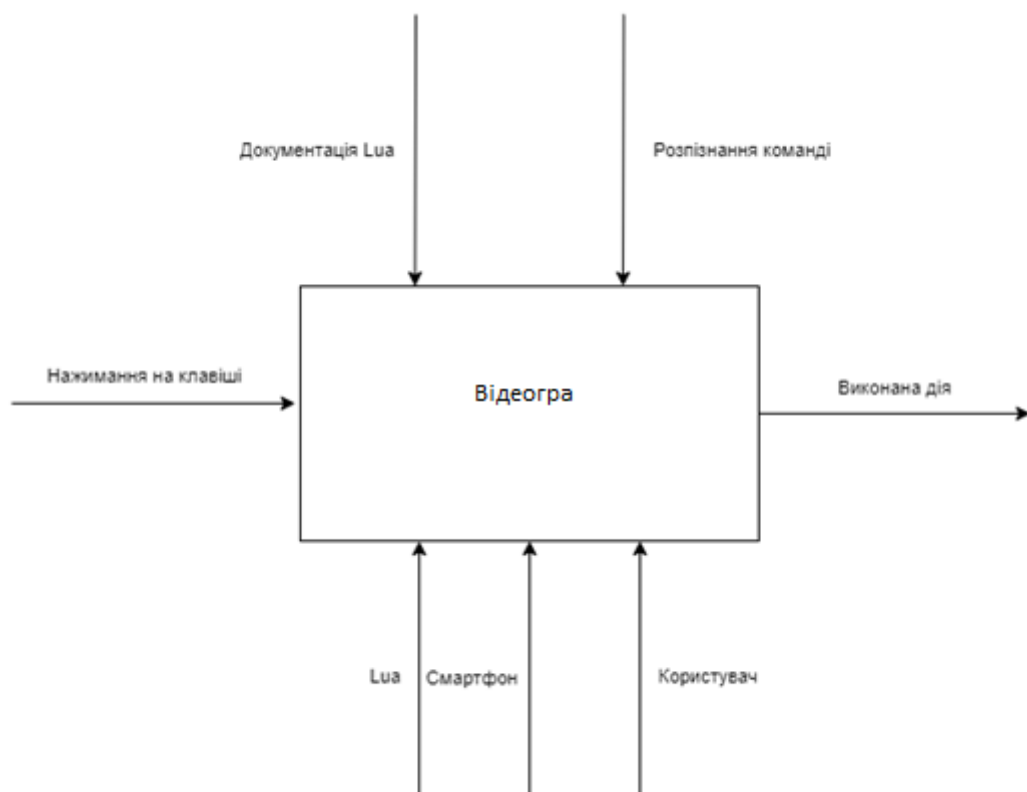


Рисунок 2.1 – Діаграма декомпозиція проєкту

Тому, у програмі будуть реалізовані такі функції як керування персонажем  
Функція керування персонажем буде це одна з найголовніших функцій.  
Завдання зміна координатів персонажа.

### 2.3 Просування алгоритму роботи системи

Механіка алгоритмів є доволі простою, після натискання гравцем відповідної точки на екрані, здійснюється відповідна анімація, рух та дія. Автоматичне створення генерація світу реалізовано за допомогою функції “random”. За межами екрану у відеогрі випадковим чином з’являтимуться перешкоди.

Створенням перешкод буде займатись відповідна функція, дана функція буде запускатися регулярно, відповідно до ігрового циклу, вона викликатиметься повторно, для обробки функціональних можливостей.

Оскільки на екрані кількість перешкод буде збільшуватись, тоді знадобиться метод відстеження. Для стеження за перешкодами буде використовуватись вбудована команда у середовище Solar 2D - “table.insert()”. Table.insert() - додає значення у таблицю, коли вказано позицію, додає значення перед елементом, що знаходиться на позиції у даний момент.

Перешкоди будуть виходити з правої частини екрану, а саме зверху або знизу та справа, вірус не буде генеруватися зліва, оскільки користувачу буде складно ухилятися, чи здійснювати інші дії.

Тепер, враховуючи дані можливі точки, програма буде генерувати випадкове число від одного до трьох.

Рух вірусів буде відбуватись через “раптовий поштовх”, завдяки такому поштовху об’єкти будуть рухатися у послідовному, стабільному напрямку. Щоб змусити вірус обертатись, пересуватись у просторі, використовуємо випадкову величину при генерації нових об’єктів.

					ДП. КН 21.434.13.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Механіка стрільби. Більшість цього алгоритму має бути простим:

- Створюємо новий об'єкт, фаєрбол у середині групи з текстурами.
- Додаємо об'єкту фізичні властивості.
- Вказуємо, що варто розглядати об'єкт як кулю. Завдяки цьому об'єкт постійно виявлятиме нові зіткнення.
- Присвоюємо фаєрболу координати гравця, для можливості знищувати перешкоди з різноманітних точок на карті.

Для розміщення фаєрболу як об'єкт, використовується статичне положення гравця, оскільки користувач рухатиметься у різноманітні сторони, це буде оптимальним рішенням.

Також такий метод дозволить правильно розташувати зброю, як по вертикалі так і по горизонталі.

На рисунку 2.2 зображена DFD діаграма.

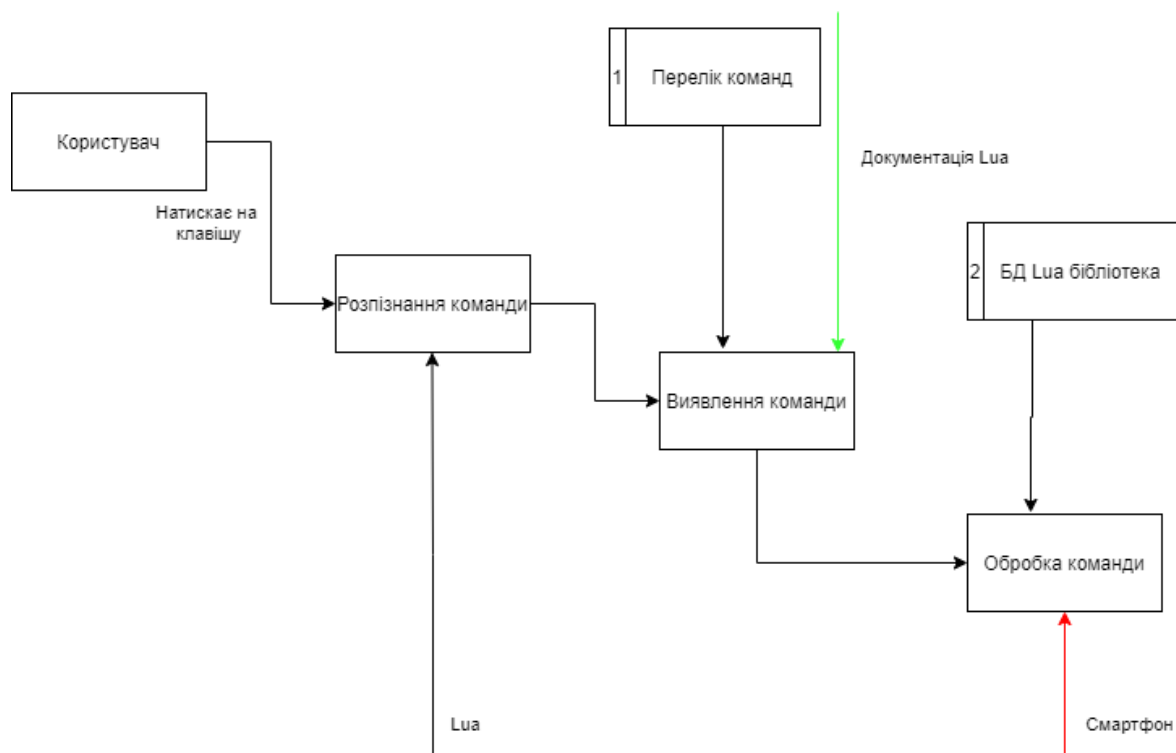


Рисунок 2.2 – DFD діаграма

На наступному рисунку, а саме, на рисунку 2.3 зображено продовження DFD діаграми.

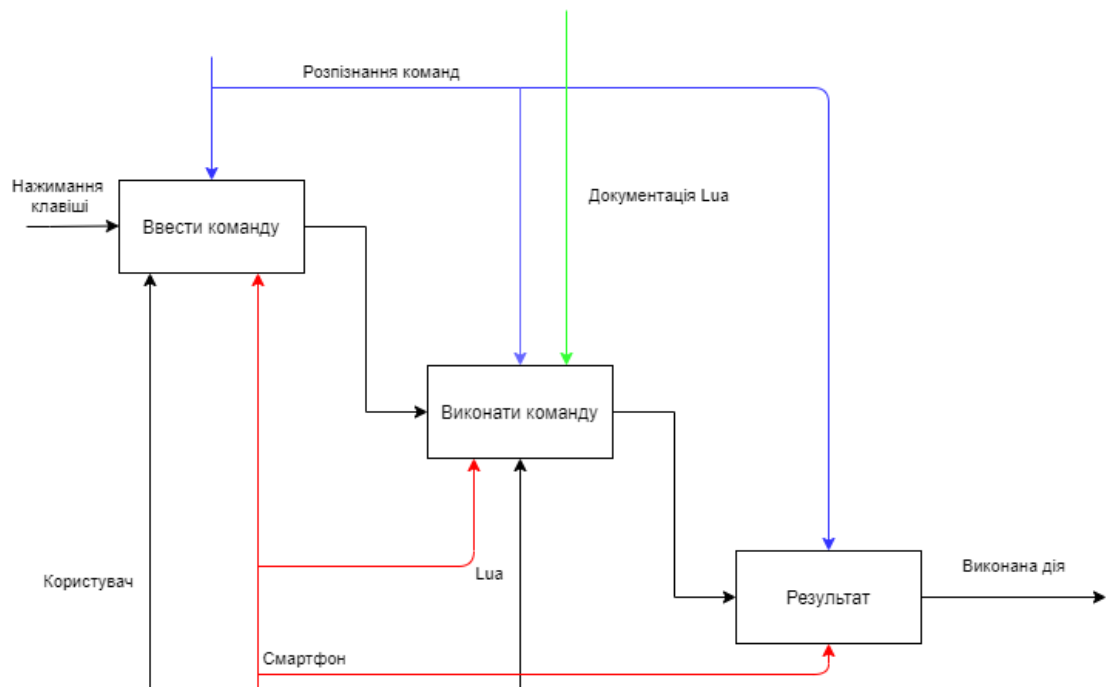


Рисунок 2.3 – продовження DFD діаграми

У вигляді кульмінації ми отримуємо хороші алгоритм для роботи відеогри. Механіка реалізована у вигляді зручних для подальших модифікацій алгоритмах. Завдяки даній реалізації додаток може отримувати оновлення системи від різноманітних користувачів.

### 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

#### 3.1 Аналіз інструментальних засобів для реалізації гри

В основному інструментальні засоби поділяються на три групи()

Конструктор відеоігор – це програма, редактор, використовує принцип роботи WYSIWYG. Конструктори переважно мають редактор рівнів, що значно спрощує наповнення додатка контентом [8].

Фреймворк – це програми, що полегшують створення технічно складних продуктів. Фреймворк містить базові компоненти, що реалізуються розробником на їх основі. Завдяки чому досягається висока продуктивність, та швидкість розробки [9].

Ігровий рушій – це спеціальні середовища розробки, ядро, що забезпечує основні технології для відеоігор, різноманітних додатків з графікою, та інших програм.

Рушій завдяки своєму ядру значно спрощує створення додатків, а багатоплатформність дозволяє програмі запускатись на різних операційних системах, таких як Android, IOS, Windows та інших [10].

#### 3.2 Вибір засобів реалізації

Додаток буде розроблятися за допомогою ігрового рушія або фреймворка.

Було проаналізовано кілька середовищ для розробки, а саме Unity, Solar 2D, Defold Engine. Переваги та недоліки середовищ:

Solar 2D – це фреймворк з відкритим кодом. Дане середовище було розроблено компанією Corona Labs inc., на рисунку 3.1 зображено логотип Solar 2D.

Фреймворк зручний у користуванні та використовує мову програмування Lua.



Рисунок 3.1 – Логотип Solar 2D

Перевагами Solar 2D є:

- Безкоштовний.
- Легкість використання.
- Багатоплатформність.
- Хороший емулятор.
- Проста у освоєнні мова програмування.

Недоліками гри є:

- Мало документації.
- Немає редактора рівнів.
- Немає розділення для коду і дизайну.

Unity – це інструмент для створення різноманітних застосунків, таких як відеігри, програми, тощо для мобільних і комп’ютерних систем, тощо.

На рисунку 3.2 зображено логотип платформера.



Рисунок 3.2 – Логотип Unity

Перевагами Unity є:

- Безкоштовний для комерційного використання.
- Багатоплатформність .
- Asset Store ( Магазин активів ).
- Можливість тестування у реальному часі.
- Велика кількість документації.

Недоліками є:

- Водяний знак при вмиканні програми.
- Великий розмір файлів.
- Погана архітектура.

Defold – це відносно нове середовище, воно набирає все більше і більше нових розробників. При розробці автори використовували кращі риси таких середовищ як Unity, Solar2D, тощо. Завдяки чому получився дивовижний “движок” для розробки програм та відеоігор.



Рисунок 3.3 – Логотип Defold

Перевагами Defold є:

- Безкоштовний.
- Легкість використання.
- Багатоплатформність.
- Хороший емулятор.
- Проста у освоєнні мова програмування.

Недоліками гри є:

- Мало документації.
- Відсутній емулятор.

Проаналізувавши переваги та недоліки різноманітних рушіїв, було вирішено розробляти проєкт на Solar 2D, вагомою перевагою був хороший емулятор. Завдяки чому, можна зразу переглянути і перевірити роботу своєї програми, емулятор зображений на рисунку 3.4



Рисунок 3.4 – Емулятор Solar2D

### 3.3 Вибір допоміжного програмного забезпечення

Для успішної розробки відеогри необхідні наступні ресурси, а саме графічний матеріал. Графічний матеріал, а саме текстури буде взято з сайту Pexels.com. Даний сайт надає безкоштовні зображення для комерційного використання та будь-якого іншого.

### 3.4 Створення гри

Перед розробкою слід створити проєктний документ (GDD). Цей документ допоможе оперативно просуватись в розробці програми (таблиця 3.1):

Таблиця 3.1 - проєктний документ

Назва гри	HiZ Arcade
Опис	Ухиляйтесь персонажем від вірусу, знищуйте по мірі проходження.
Елементи управління	Натисніть на персонажа для знищення перешкод; перетягуйте персонажа для переіщення вправо та вліво.
Звук	Звук вогню, при зіткненні з перешкодами

У Solar 2D створюємо новий проєкт, назва проєкту “HiZ Arcade”, розширення екрану - Tablet Preset, що містить робоче розширення 1024x768р. Меню налаштувань Solar 2D зображено на рисунку 3.5

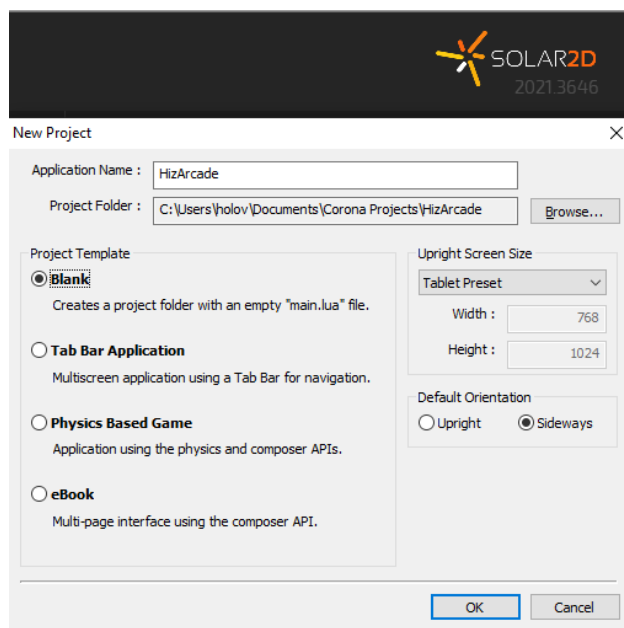


Рисунок 3.5 – Налаштування проєкту

Текстури для відеогри використовуються одним файлом, завдяки такому використанню, додаток витрачає менше ресурсів, що добре впливає на акумулятор пристрою, збільшуючи його автономну роботу. На рисунку 3.6 зображені ігрові об'єкти.

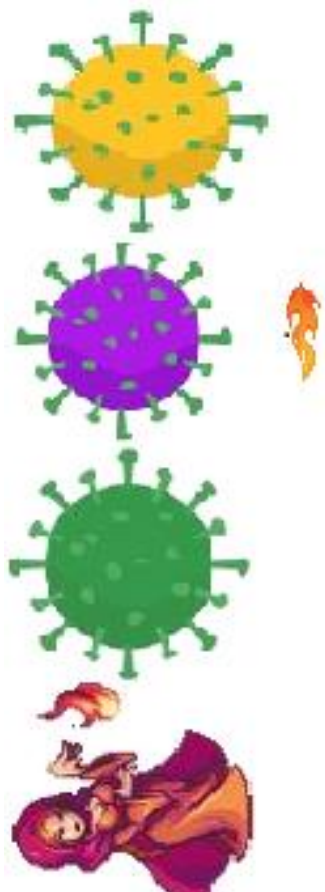


Рисунок 3.6 – Ігрові об'єкти

Відеогра використовуватиме фізичний движок, для оперативної роботи, він оголошується в першу чергу. Реалізація:

```
Local phys = require("physics")
phys.start()
phys.setGravity ( 0, 0 )
```

За замовчуванням середовище імітує звичайне земне тяжіння, що призводить до падіння об'єктів до нижньої частини екрану.

Команда `phys.setGravity ( x, y )`, яку ми використовуємо може відтворювати гравітацію у вертикальному, та горизонтальних напрямках у та x. Для зручності

					ДП. КН 21.434.13.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

кірування персонажем, гравітація була виставлена на 0, тому гравітація на гравця не діє. Противник буде рухатись хаотично. Як сказано вище, за замовчуванням середовище імітує звичайне земне тяжіння. Для перешкоди були створенні окремі налаштування фізики:

```
physics.addBody ( virus, "dynamic", {radius=35, bounce=0.7} )
```

radius - це радіус взаємодії з іншими об'єктами. bounce - властивість перешкоди, значення регулює інерцію від зіткнень, більше значення - більша інерція.

Пересування персонажем реалізовано через дотик, touch. Спочатку створювалась спеціальна клавіатура, але згодом було спрощено до звичайного дотику:

```
player.addEventListener("touch", ds )
```

Для знищення об'єктів, використовується наступна функція:

```
Local function fireball()  
Local nFire = display.newImageRect(localGroup, texture, 5, 14,  
40)  
phys.addBody( nFire, "dynamic", {isSensor=true})  
  
nFire.isBullet = true  
nFire.x = player.x  
nFire.y = player.y  
nFire:toBack()  
End
```

Реалізовуємо снаряд як об'єкт типу датчика isSensor. IsBullet - вказує, що об'єкт слід розглядати як "кулю". Завдяки цьому об'єкт постійно виявляє зіткнення. Запуск функції здійснюється за допомогою звичайного натиску:

```
player.addEventListener( "tap", fireball )
```

Перша умова зіткнення: фаєрбол і вірус. Властивість myName, яку було присвоєно кожному об'єкту. Використовується як засіб виявлення зіткнень між об'єктами.

					ДП. КН 21.434.13.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

Оскільки генерація вірусу відбувається хаотично, типи об'єктів можуть зіткнутися у різних послідовностях, таких як фаєрбол вірус, або вірус фаєрбол, користувач вірус та вірус користувач:

```
If ((o1.myName == "fire" and o2.myName == "virus") or (o2.myName == "virus" and o1.myName == "fire"))
```

```
Then
```

```
display.remove(o1)
```

```
display.remove(o2)
```

Після виконання цієї функції гравець отримує нагороду, а саме збільшення score на 100 одиниць:

```
point = point + 100
```

```
Pointt.text = "Score:"..point
```

Другою умовою зіткнення буде користувач та вірус:

```
If ((o1.myName == "player" and o2.myName == "virus") or (o2.myName == "virus" and o1.myName == "player"))
```

```
Then
```

```
If (died == false) then died = true
```

Після зіткнення з перешкодою гравець отримує покарання, зменшення кількості спроб для проходження, програма повинна переконатись, що у гравця достатня кількість спроб для проходження далі, якщо спроб більше немає гру потрібно завершити:

```
lives = hp - 1
```

```
livesText.text = "Lives:"..hp
```

```
if ( hp == 0 ) then
```

```
display.remove(player)
```

Настройки конфігурації додатка, а саме масштабування вмісту. Якщо програма не буде відповідати співвідношенню сторін, наприклад 16:9 або 4:3. Letterbox або zoomEven - це два найпопулярніші варіанти налаштування розширення. У програмі використовується zoomEven, саме він масштабує область вмісту, щоб “залити” екран. Завдяки “залиттю”, програма зберігає своє співвідношення сторін, а весь вміст розміщений на екрані.

					ДП. КН 21.434.13.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.5 Вибір способу захисту гри

Захист додатка реалізовано KeyStore Explorer 5.4.4 - це проста і швидка у користуванні програма, у даній програмі містяться пакети для усіх популярних операційних систем. Програма використовує графічний інтерфейс замість командного рядка. Завдяки KeyStore Explorer спрощує такі завдання як підписання коду та генерацію пари ключів.

Після запуску KeyStore Explorer, було створенно новий ключ Java KeyStore – рисунок 3.7.

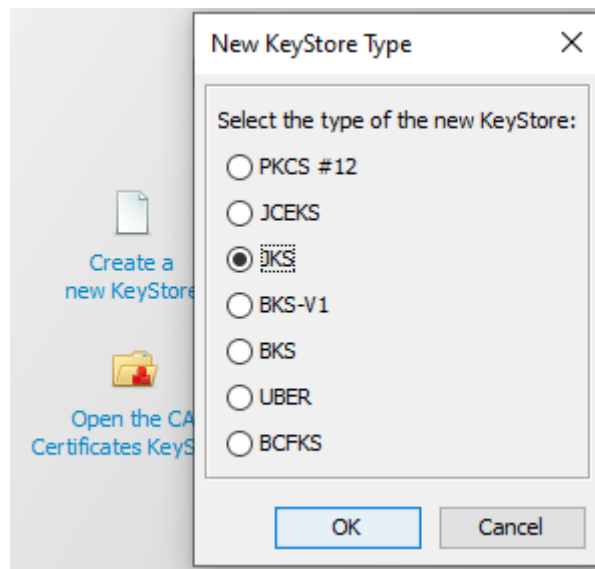


Рисунок 3.7 – Створення ключа JKS

Створення нової пари ключів з алгоритмом RSA зображений на рисунку 3.8.

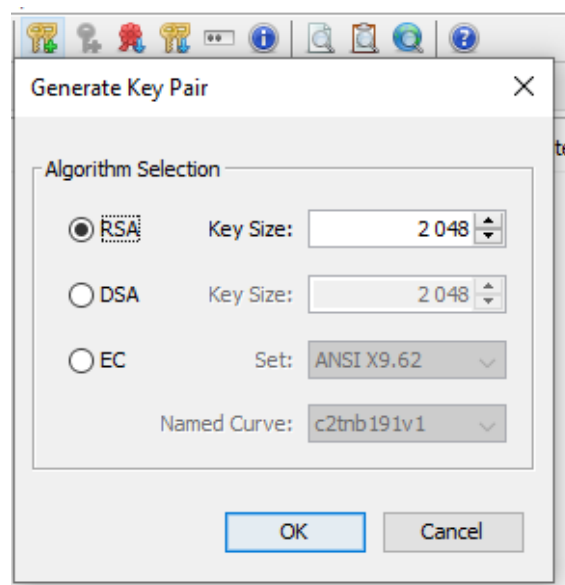


Рисунок 3.8 – Створення нової пари ключів

Алгоритмом підпису було обрано SHA-256 з RSA, строк дії було змінено на 7 років. Рисунок 3.9.

Рисунок 3.9 – Алгоритм підпису SHA-256

На рисунку 3.10 зображено заповнення даними про компанію, тощо.

Рисунок 3.10 – Інформація про компанію

Після заповнення інформації про компанію, програма просить придумати псевдонім. Рисунок 3.11, псевдонім: MW8TD, з часом він знадобиться.

					ДП. КН 21.434.13.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

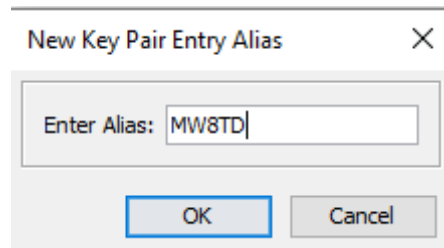


Рисунок 3.11 – Псевдонім

Завдяки проведеним операціям, у сховищі ключів можна побачити новий ключ, його можна ідентифікувати за псевдонімом, рисунок 3.12. Залишилось зберегти ключ на персональному комп'ютері з метою а подальшому його використанні.

Untitled-1 \* %







			Entry Name	Algorithm	Key Size	Certificate Expiry	Last Modified
			mw8td	RSA	2048	09.06.2022 23:30:35...	09.06.2021 23:36:28...

Рисунок 3.12 – JKS ключ

На рисунку зображено процес компіляції, де ми вже завантажили відповідний ключ .Під час компіляції програми, у рядок Keystore завантажуюмо уже створений JKS ключ – рисунок 3.13.

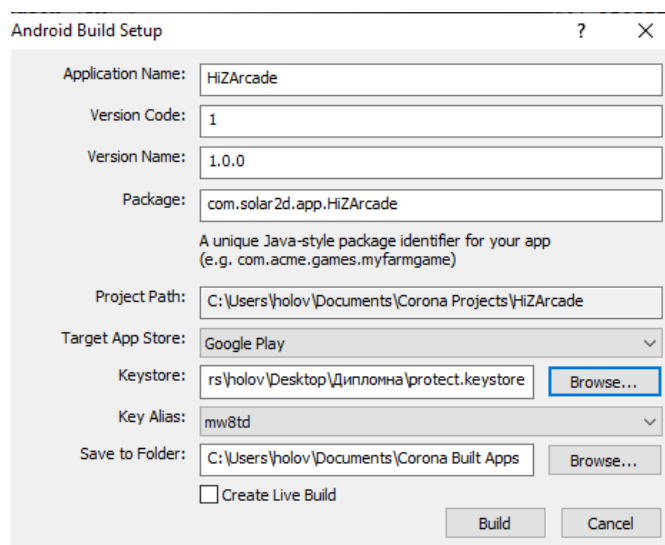


Рисунок 3.12 – Компіляція проєкту з JKS

### 3.6 Тестування

Після завершення роботи над функціоналом, механікою та контентом відеогри, починається останній етап - тестування.

Запуск відеогри відбувається за допомогою емулятора Solar 2D – рисунок 3.13

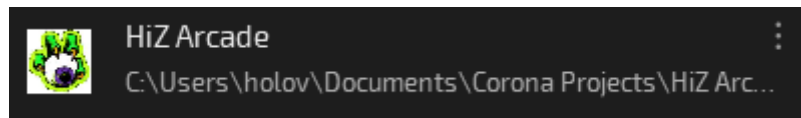


Рисунок 3.13 – Ярлик відеогри

Після запуску відеогри, користувача зустрічає головне меню, на якому зображений логотип відеогри.– Рисунок 3.14.

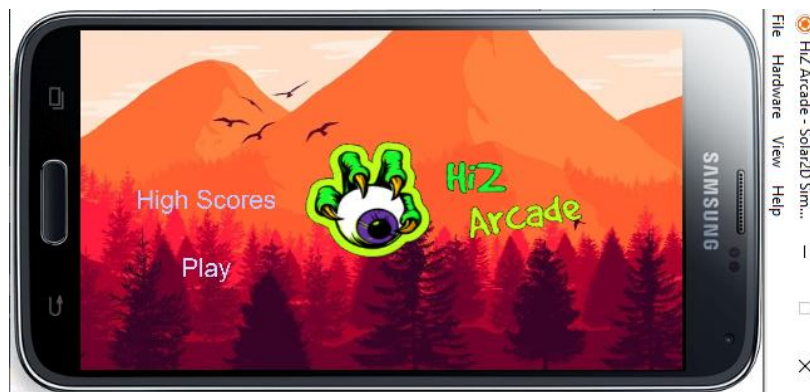


Рисунок 3.14 – Головне меню

Після того як користувач натисне на кнопку “Play”- запуститься відеогра.

Одразу ж після старту гравця, що зображено на рисунку 3.15 зустрічають декілька противників, яких він без проблем може знищити.



Рисунок 3.15 – Запуск відеогри

Змн.	Арк.	№ докум.	Підпис	Дата

ДП. КН 21.434.13.000 ПЗ

Арк.

32

Керування персонажем реалізовано без перешкод. Персонаж вільно може пересуватись в різноманітні сторони, це зображено на рисунку 3.16.



Рисунок 3.16 – Пересування персонажа

На рисунку 3.17 відображається кількість здобутих очків за знищення мішеней та кількість життів гравця відображається у правому куті екрана.



Рисунок 3.17 – Кількість мішеней та життів

Фішкою відеогри є вертикальна таблиця рекордів, завдяки чому на екрані вдається помістити значно більше інформації. Після того як гравець зазнав невдачі, на екрані з'явиться нове меню High Score, на якому будуть зображені кращі ігри користувача – це зображено на рисунок 3.18.

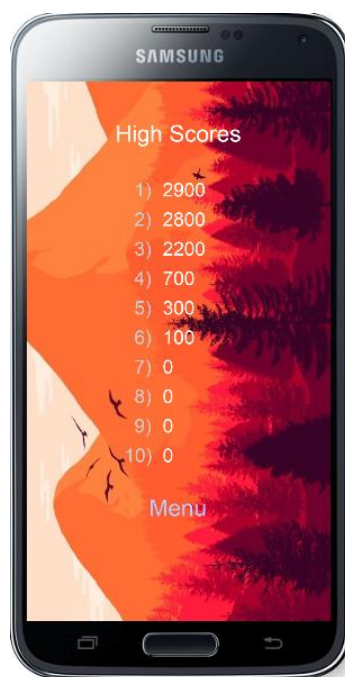


Рисунок 3.18 – Статистика ігор користувача

Під час тестування додатку, його робота була стабільною. Протягом усього тестування, лаги та баги були відсутніми, функції програми виконуються без будь яких затримок. Різноманітні фактори такі як дзвінки, тощо на пошкодження ігрового процесу не впливають, додаток зупиняється.

## 4 ТЕХНІЧНО ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

### 4.1 Аналіз ринку

Основні техніко–економічні та експлуатаційні характеристики нового виробу.

Додаток орієнтований на смартфони з роздільною здатністю HD 720p, це не значить, що на меншому розширенні додаток не запуститься, розширення регулюється за допомогою “ZoomEven” - масштабу область вмісту, щоб заповнити екран, зберігаючи однакове співвідношення сторін. більша роздільна здатність забезпечує кращу передачу кольору, чіткість зображення.

Для успішного запуску потрібна версія “android” не нижче 4.4. Гра займає 83 мегабайтів. Якщо системних характеристик смартфона не вистачає, програма не встановлюється, у разі браку інших ресурсів, таких як RAM, користувач відчуватиме дискомфорт у вигляді низької частоти кадрів та невеликих фризів, у разі критичного браку ресурсів, додаток закриватиметься видаючи помилку.

На ринку уже існують аналогічні продукти. Додаток орієнтований на потенційних замовників, а саме дітей та дорослих. Дослідження IT-ринку показує, що одним з стабільніших, оптимізованих та безпечних ринків є “Play Store”, також можлива реалізація товару на таких ринках як “solar 2d”, “defold engine” та інші. За тиждень очікується щонайменше 12 проданих копій, за місяць 50. Сервісне обслуговування, допродажне та післяпродажне імовірно буде реалізовано через оновлення додатку. Головними конкурентом на ринку є відеогра “Vector” компанії Nekki – Action and Fighting Games. Даний проєкт має стабільну аудиторію та є не мало популярним на ринку.

Рівень цін на продукцію конкурентів, додаток “Vector” коштує більше 21 гриві, має товари на суму від 10,66 до 53,64 гривень.

					ДП. КН 21.434.13.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

## 4.2 Розрахування витрат на розробку проекту

У таблиці 4.1 поданий кошторис.

Таблиця 4.1 – Кошторис

Найменування статей витрат	Сума, грн	Обґрунтування
1.Зарплата проектувальників	5 000	При розробці відеогри потрібно розробити функціонал і відповідно протестувати
2.Відрахування на соціальні потреби	1 1000	ЄСВ – 22%
3.Контрагентські роботи і послуги	0	
4.Витрати на відрядження	0	
5.Інші витрати	1 000	5 000 – 80%
6.Усього прямих витрат	7 600	5 500 + 1 100 + 1 000
7.Накладні витрати	1 520	7 600 – 80%
8.Планові накопичення	1 824	(7 600 + 1 520) - 80%
9.Усього кошторисна вартість проекту	7 924	5 000 + 1 000 + 1 824
10.Податок на додану вартість	2 377	7 924 – 70%
11.Загалом договірна ціна розробки	22 376	7 924 + 2 377 + 12 075

Заробітна плата працівникам розраховується на двох програмістів.  
Рисунок 4.2 розрахунок заробітної плати.

У таблиці 4.2 подано розрахунок заробітної плати.

Таблиця 4.2 – Розрахунок заробітної плати

№	Посада виконавця	Оклад грн/міс	Відрахування грн/міс	Кількість		Сума з/п, грн.
				Чол	Місяців	
1	Технік- програміст	5 000	975	2	3	4 025
Усього зарплати:						12 075

#### 4.3 Обґрунтування необхідності розробки

У економічній частині дипломного проєкту обґрунтовано доцільність розробки програмного продукту, розраховано вартість на розробку та визначено договірну ціну (ринкову), яка складає 22 376 гривень. Розрахований період протягом якого було реалізовано розробку. Проаналізувавши характеристики нового програмного продукту та аналогу, виявлено, що за кількістю функцій розроблена відеогра не відстає від ринкового аналогу.

При виконанні економічної частини, зробивши усі необхідні розрахунки, виявлено, що технологія розробки відповідає оптимальному рівню витрат, у кінцевому підсумку є розроблений продукт який є конкурентоспроможним на ринку.

## ВИСНОВКИ

Використовуючи різноманітні інструментальні засоби, була розроблена відеогра. На Solar 2D розробка пройшла в швидкому темпі, інтуїтивний та зручний інтерфейс емулятора позитивно впливає на розробку. На самому фреймворку було написано код та усі елементи відеогри, усе було написано вручну. Solar 2D надає свободу для реалізації, стимулює та оптимізовує продуктивність роботи. Протягом програмування використовувалась мова Lua – це скриптова мова, вона компактна і швидка. Основним мінусом цієї мови була відсутня документація, вірніше мала її кількість і коли виникала проблема з кодом навіть незначна витрачалось багато часу на її усунення.

Протягом розробки відеогри було засвоєно теоретичні відомості, про сучасні засоби розробки. Були переглянуті методи побудови відеоігор та системи системи розробки. Була реалізована ігрова локація, система керування персонажа, можливість знищення об'єктів та взаємодія з об'єктами.

					<i>ДП. КН 21.434.13.000 ПЗ</i>	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Аркада (гра). *Wikipedia*: вебсайт. URL:  
[https://uk.wikipedia.org/wiki/Аркада\\_\(гра\)](https://uk.wikipedia.org/wiki/Аркада_(гра)) (дата звернення: 10.02.2021).
2. Android. *Wikipedia*: вебсайт. URL:  
<https://uk.wikipedia.org/wiki/Android> (дата звернення: 15.02.2021).
3. История версий Android. *Wikipedia*: вебсайт. URL:  
[https://ru.wikipedia.org/wiki/История\\_версий\\_Android](https://ru.wikipedia.org/wiki/История_версий_Android) (дата звернення: 10.02.2021).
4. Vector (гра). *GooglePlay*: вебсайт. URL:  
<https://play.google.com/store/apps/details?id=com.nekki.vector&hl=uk&gl=US> (дата звернення: 10.02.2021).
5. Sonic the Hedgehog. *Wikipedia*: вебсайт. URL:  
[https://en.wikipedia.org/wiki/Sonic\\_the\\_Hedgehog](https://en.wikipedia.org/wiki/Sonic_the_Hedgehog) (дата звернення: 10.02.2021).
6. Fast like a Fox. *GooglePlay*: вебсайт. URL:  
<https://play.google.com/store/apps/details?id=com.waybefore.fastlikeafox&hl=uk&gl=US> (дата звернення: 10.02.2021).
7. Super Bob`s World 2020. *GooglePlay*: вебсайт. URL:  
<https://play.google.com/store/apps/details?id=com.os.falcon.bob2.adventure.world&hl=uk&gl=US> (дата звернення: 10.02.2021).
8. Конструктор відеоігор. *Wikipedia*: вебсайт. URL:  
[https://uk.wikipedia.org/wiki/конструктор\\_відеоігор](https://uk.wikipedia.org/wiki/конструктор_відеоігор) (дата звернення: 10.02.2021).
9. Фреймворк. *Wikipedia*: вебсайт. URL:  
<https://ru.wikipedia.org/wiki/фреймворк> (дата звернення: 10.02.2021).
10. Ігровий рушій. *Wikipedia*: вебсайт. URL:  
[https://uk.wikipedia.org/wiki/ігровий\\_рушій](https://uk.wikipedia.org/wiki/ігровий_рушій) (дата звернення: 10.02.2021).

## ДОДАТКИ

### Додаток А

#### Програмний код відеогри

```
local composer = require( "composer" )
local scene = composer.newScene()
local physics = require( "physics" )
physics.start()
physics.setGravity( 0, 0 )

local sheetOptions =
{
    frames =
    {
        {
            x = 0,
            y = 0,
            width = 102,
            height = 85
        },
        {
            x = 0,
            y = 85,
            width = 90,
            height = 83
        },
        {
            x = 0,
            y = 168,
            width = 100,
            height = 97
        },
    },
}
```

					ДП. КН 21.434.13.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        x = 0,
        y = 265,
        width = 98,
        height = 79
    },
    {
        x = 98,
        y = 265,
        width = 14,
        height = 40
    },
},
}

local objectSheet = graphics.newImageSheet( "gameObjects.png"
, sheetOptions )

local lives = 3
local score = 0
local died = false

local asteroidsTable = {}

local ship
local gameLoopTimer
local livesText
local scoreText

local backGroup
local mainGroup
local uiGroup

local explosionSound
local fireSound
local musicTrack

```

```

local function updateText()
    livesText.text = "Lives: " .. lives
    scoreText.text = "Score: " .. score
end

local function createAsteroid()

    local newAsteroid = display.newImageRect( mainGroup, obje
ctSheet, 1, 102, 85 )
    table.insert( asteroidsTable, newAsteroid )
    physics.addBody( newAsteroid, "dynamic", { radius=40, bou
nce=0.8 } )
    newAsteroid.myName = "asteroid"

    local whereFrom = math.random( 3 )

    if ( whereFrom == 1 ) then

        newAsteroid.x = -60
        newAsteroid.y = math.random( 500 )
        newAsteroid:setLinearVelocity( math.random( 40,120 ),
math.random( 20,60 ) )
    elseif ( whereFrom == 2 ) then

        newAsteroid.x = math.random( display.contentWidth )
        newAsteroid.y = -60
        newAsteroid:setLinearVelocity( math.random( -
40,40 ), math.random( 40,120 ) )
    elseif ( whereFrom == 3 ) then

        newAsteroid.x = display.contentWidth + 60
        newAsteroid.y = math.random( 500 )
        newAsteroid:setLinearVelocity( math.random( -120,-
40 ), math.random( 20,60 ) )
    end
end

```

					ДП. КН 21.434.13.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        newAsteroid:applyTorque( math.random( -6,6 ) )
end

local function fireLaser()

    audio.play( fireSound )

    local newLaser = display.newImageRect( mainGroup, objectSheet, 5, 14, 40 )
    physics.addBody( newLaser, "dynamic", { isSensor=true } )
    newLaser.isBullet = true
    newLaser.myName = "laser"

    newLaser.x = ship.x
    newLaser.y = ship.y
    newLaser:toBack()

    transition.to( newLaser, { y=-40, time=500,
        onComplete = function() display.remove( newLaser ) end
    } )
end

local function dragShip( event )

    local ship = event.target
    local phase = event.phase

    if ( "began" == phase ) then

        display.currentStage:setFocus( ship )

        ship.touchOffsetX = event.x - ship.x

```

```

elseif ( "moved" == phase ) then

    ship.x = event.x - ship.touchOffsetX

elseif ( "ended" == phase or "cancelled" == phase ) then

    display.currentStage:setFocus( nil )
end

return true
end

local function gameLoop()

    createAsteroid()

    for i = #asteroidsTable, 1, -1 do
        local thisAsteroid = asteroidsTable[i]

        if ( thisAsteroid.x < -100 or
            thisAsteroid.x > display.contentWidth + 100 or
            thisAsteroid.y < -100 or
            thisAsteroid.y > display.contentHeight + 100 )
        then
            display.remove( thisAsteroid )
            table.remove( asteroidsTable, i )
        end
    end
end

local function restoreShip()

    ship.isBodyActive = false
    ship.x = display.contentCenterX

```

```

    ship.y = display.contentHeight - 100

    transition.to( ship, { alpha=1, time=4000,
        onComplete = function()
            ship.isBodyActive = true
            died = false
        end
    } )
end

local function endGame()
    composer.setVariable( "finalScore", score )
    composer.gotoScene( "highscores", { time=800, effect="crossFade" } )
end

local function onCollision( event )

    if ( event.phase == "began" ) then

        local obj1 = event.object1
        local obj2 = event.object2

        if ( ( obj1.myName == "laser" and obj2.myName == "asteroid" ) or
            ( obj1.myName == "asteroid" and obj2.myName == "laser" ) )
        then

            display.remove( obj1 )
            display.remove( obj2 )

            audio.play( explosionSound )

```

```

        for i = #asteroidsTable, 1, -1 do
            if ( asteroidsTable[i] == obj1 or asteroidsTable[i] == obj2 ) then
                table.remove( asteroidsTable, i )
                break
            end
        end

        score = score + 100
        scoreText.text = "Score: " .. score

        elseif ( ( obj1.myName == "ship" and obj2.myName == "asteroid" ) or
            ( obj1.myName == "asteroid" and obj2.myName == "ship" ) )
        then
            if ( died == false ) then
                died = true

                audio.play( explosionSound )

                lives = lives - 1
                livesText.text = "Lives: " .. lives

                if ( lives == 0 ) then
                    display.remove( ship )
                    timer.performWithDelay( 2000, endGame )
                else
                    ship.alpha = 0
                    timer.performWithDelay( 1000, restoreShip
                )
            end
        end

```

					ДП. КН 21.434.13.000 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        end
    end
end
end

function scene:create( event )

    local sceneGroup = self.view

    physics.pause()

    backGroup = display.newGroup()
    sceneGroup:insert( backGroup )
    mainGroup = display.newGroup()
    sceneGroup:insert( mainGroup )

    uiGroup = display.newGroup()
    sceneGroup:insert( uiGroup )

    local background = display.newImageRect( backGroup, "back
ground.png", 800, 1400 )
    background.x = display.contentCenterX
    background.y = display.contentCenterY

    ship = display.newImageRect( mainGroup, objectSheet, 4, 9
8, 79 )
    ship.x = display.contentCenterX
    ship.y = display.contentHeight - 100
    physics.addBody( ship, { radius=30, isSensor=true } )
    ship.myName = "ship"

    livesText = display.newText( uiGroup, "Lives: " .. lives,
200, 80, native.systemFont, 36 )

```

					ДП. КН 21.434.13.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        scoreText = display.newText( uiGroup, "Score: " .. score,
400, 80, native.systemFont, 36 )

        ship:addEventListener( "tap", fireLaser )
        ship:addEventListener( "touch", dragShip )

        explosionSound = audio.loadSound( "audio/explosion.wav" )
        fireSound = audio.loadSound( "audio/fire.wav" )
        musicTrack = audio.loadStream( "audio/80s-Space-
Game_Looping.wav" )
    end

function scene:show( event )

    local sceneGroup = self.view
    local phase = event.phase

    if ( phase == "will" ) then

        elseif ( phase == "did" ) then

            physics.start()
            Runtime:addEventListener( "collision", onCollision )
            gameLoopTimer = timer.performWithDelay( 500, gameLoop
, 0 )

            audio.play( musicTrack, { channel=1, loops=-1 } )
        end
    end

function scene:hide( event )
    local sceneGroup = self.view
    local phase = event.phase
    if ( phase == "will" ) then

```

```

        timer.cancel( gameLoopTimer )
    elseif ( phase == "did" ) then
        Runtime:removeEventListener( "collision", onCollision
    )

        physics.pause()

        audio.stop( 1 )
        composer.removeScene( "game" )
    end
end

function scene:destroy( event )
    local sceneGroup = self.view
    audio.dispose( explosionSound )
    audio.dispose( fireSound )
    audio.dispose( musicTrack )
end

scene:addEventListener( "create", scene )
scene:addEventListener( "show", scene )
scene:addEventListener( "hide", scene )
scene:addEventListener( "destroy", scene )
return scene

```