

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 "Пригода"»

Студент групи К-47

Буднік Т.А.

\_\_\_\_\_  
(підпис)

Керівник проєкту

Кузик В.П.

\_\_\_\_\_  
(підпис)

Консультанти:

з техніко-економічного  
обґрунтування

Меленчук Л.І.

\_\_\_\_\_  
(підпис)

нормоконтролер

Кульчинська Н.З.

\_\_\_\_\_  
(підпис)

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 проектування  
на здобуття освітньо-кваліфікаційного рівня «молодший спеціаліст»  
студенту Будніку Тарасу Анатолійовичу  
(прізвище, ім'я та по-батькові студента)

1. Тема проєкту \_\_\_\_\_

\_\_\_\_\_

затверджена 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 проєкту: \_\_\_\_\_

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

### КАЛЕНДАРНИЙ ПЛАН дипломного проєктування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до дипломного проєктування	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 проєкту 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) 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 проєкту 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 проєкту, доопрацювання	11.06.21	
13.	Підготовка до захисту дипломного проєкту	18.06.21	22.06.21
14.	Захист дипломного проєкту	23.06.21	

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

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

## Реферат

Ігровий мобільний додаток в жанрі пригода. Дипломний проєкт. Буднік Тарас Анатолійович. Галицький коледж імені В'ячеслава Чорновола, відділення комп'ютерних та видавничих технологій. Спеціальність 122 «Комп'ютерні науки». ГК, 2021. Сторінок – 53, рисунків – 24, додатків – 1.

Дипломний проєкт присвячений ігровому мобільному додатку, його аналізу та розробці.

У першому розділі було проведено аналіз схожих ігрових мобільних додатків, розглянуто декілька існуючих ігор та описано їх.

У другому розділі здійснено формалізацію вимог до системи, а саме – вимог до програмного продукту, інтерфейсу користувача та бази даних. Виконано проєктування системи та її складових, а також описано алгоритм її функціонування.

Третій розділ розкриває технології та засоби реалізації, містить опис розробки і тестування основних функцій системи та інтерфейсу.

В четвертому розділі проводиться техніко-економічне обґрунтування.

Об'єкт дослідження – процес розробки ігрових мобільних додатків.

Метою проєкту є реалізація ігрового мобільного додатку в жанрі «Пригода».

Результатом роботи над проєктом є гра.

МУЛЬТИМЕДІА, ІГРИ, ПРОЄКТ, ПРИГОДА, МОБІЛЬНИЙ ДОДАТОК, СИСТЕМА, АЛГОРИТМ, ГЕЙМПЛЕЙ, ГРАВЕЦЬ, ДИЗАЙН, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, КЛЮЧ.

## Abstract

Gaming mobile application in the adventure genre. Diploma project. Budnik Taras Anatoliyovych. Vyacheslav Chornovil Galician College, Department of Computer and Publishing Technologies. Specialty 122 "Computer Science". GK, 2021. Pages - 54, figures - 24, appendices - 1.

Thesis is devoted to the game mobile application, its analysis and development. In the first section of the thesis, an analysis of similar game mobile applications was conducted, several existing games were considered and described.

The second section formalizes the requirements for the system, namely the requirements for the software product, user interface and database.

The system and its components are designed, and the algorithm of its functioning is described.

The third section reveals the technologies and means of implementation, contains a description of the development and testing of the main functions of the system and interface.

The fourth section provides a feasibility study.

The object of research is the process of developing mobile game applications. The purpose of the project is to implement a gaming mobile application in the genre of "Adventure".

The result of working on the project is a game.

MULTIMEDIA, GAMES, PROJECT, ADVENTURE, MOBILE APP, SYSTEM, ALGORITHM, GAMEPLAY, PLAYER, DESIGN, SOFTWARE, KEY.

## ЗМІСТ

Вступ.....	7
1 Аналіз предметної області .....	8
1.1 Актуальність теми.....	8
1.2 Аналіз наявних рішень .....	13
1.3 Постановка задачі.....	18
2 Проєктування автоматизованої продукції .....	19
2.1 Формалізація вимог до системи .....	19
2.2 Проєктування структури систем .....	22
2.3 Проєктування алгоритму роботи систем .....	23
3 Реалізація створеної системи.....	25
3.1 Аналіз інструментальних засобів для реалізації гри .....	25
3.2 Вибір засобів реалізації .....	25
3.3 Реалізація програмного забезпечення.....	29
3.4 Вибір способу захисту продукту .....	31
3.5 Тестування роботи системи .....	34
4 Техніко-економічне обґрунтування .....	36
4.1 Аналіз ринку збуту продукту чи послуги.....	36
4.2 Розрахування витрат на розробку проєкту .....	37
4.3 Обґрунтування необхідності розробки .....	38
4.4 Робіть більше, ніж просто розважайте.....	39
Висновки .....	40
Перелік джерел посилання .....	41
Додатки.....	43

					<i>ДП. КН 21.433.02.000 ПЗ</i>			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Буднік Т.А.						
Перевір.		Кузик В.М.						
Реценз.		Павлюс В.П.						
Н.контр.		Кульчинська Н.З.						
Зав. відділ.		Чубей О.О.						
					Літ.	Арк.	Аркушів	
							5	55

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ПЗ – програмне забезпечення

NPC – Non-Player Character

NPC – неігровий персонаж

RPG – role-playing game

RPG – рольова відеогра

JKS – Java KeyStore

JKS – сховище сертифікатів безпеки відкритих ключів

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

В теперішній час нереально уявити повсякденне життя без безперервного притоку інформації. Для виконання цих запитів використовуються новітні засоби мультимедіа : телебачення, Інтернет, тощо.

Мультимедіа - це форма спілкування, яка поєднує різні форми вмісту, такі як текст, аудіо, зображення, анімація чи відео.

Відеоігра - це відеоігра, яку зазвичай грають на мобільному телефоні, яка передбачає взаємодію з користувальницьким інтерфейсом або пристроєм введення - наприклад, джойстиком, контролером, клавіатурою або пристроєм, що зчитує рух - для формування візуального зворотного зв'язку.

Електронні пристрої які застосовуються для ігор - ігрові платформи. Наприклад, до таких платформ належать ПК та ігрова консоль. Пристрій введення, який використовується для того щоб керувати грою, називається ігровим контролером, наприклад: джойстик, клавіатура та мишка, геймпад або сенсорний екран телефона чи планшета. Комп'ютерні та мобільні відеоігри набули великої популярності за останній час та посіли високе місце на ринку розваг. Швидкий розвиток індустрії розробки мобільних та комп'ютерних ігор.

Для того щоб задоволити потреби розробників було створенно велику кількість інструментальних засобів та продовжується розробка нових. Загальна мета всіх інструментальних засобів - покращення та полегшення розробки, використання передових технологій обробки фізики, графіки, забезпечення кросплатформ розроблених проєктів.

Метою цієї роботи є збір теоретичних відомостей про сучасні інструментальні засоби для побудови мобільних ігор. Аналіз та огляд популярних представників програмного забезпечення для створення ігрових додатків. Постановка задач для проєктів ігор, і спрощених вимоги для сучасних ігрових проєктів. Детальний аналіз найбільш характерних представників, розробка з їх допомогою ігрових додатків. Визначення оптимальних сфер використання для інструментальних засобів спираючись на аналіз процесу розробки ігрових ПЗ.

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		



# 1 АНАЛІЗ ПРИДМЕТНОЇ ОБЛАСТІ

## 1.1 Актуальність теми

Пригодницькі ігри фокусуються на вирішенні головоломок в рамках розповіді, як правило, з невеликою кількістю елементів дії або без них. Інші популярні назви цього жанру - «графічна пригода» або «пригода вказуй і клацни», але вони представляють лише частину набагато більш широкого і різноманітного ігрового кола.

Пригодницькі ігри не засновані на тому, що словник визначає як «пригода». Деякі є, але багато хто залишає небезпеку та хвилювання для більш спокійних, вдумливих починань. Вони також не є: рольовими іграми, що передбачають великі бойові дії, побудову команди та управління очками; екшн пригоди, такі як *Uncharted* та *Prince of Persia*, де головоломки вирішення - явно вторинна увага; ігри на платформі з бічним прокруткою, такі як *Mario* або *LittleBigPlanet*; чисті головоломки, такі як *Bejeweled* або *Tetris*.

Але, ярлики можуть зайняти нас поки що. Багато ігор розширюють традиційні жанрові межі по-новому і цікаво, залишаючись пригод іграми в їх основі. *Dreamfall*, продовження точки і клацніть класичний *Найдовша подорож*, включає кілька стелсів та бойових сцен. Важкий дощ - це нова різновид інтерактивних фільмів у стилі кінофільмів, що включають управління рухом та швидкі події. На іншому кінці спектра, такі назви, як *Mystery Case Files: Dire Grove*, поєднуються в безлічі стильових полювань на сміттярів у стилі *Where's Waldo?* Навіть портал, який дає вам пістолет для вирішення головоломок на основі фізики замість того, щоб вбивати, має право називатися пригодою.

Звичайно, історії, головоломки та дослідження не обмежуються лише пригодами. Все більше ігор поза жанром включають елементи пригодницьких ігор, такі як *Scribblenauts*, *Braid* та *Limbo*. Враховуючи їх спільні функції, ми іноді розглядаємо їх як спеціальні «ігри на інтереси», хоча завжди з розумінням того, що вони виходять за рамки нашого визначення пригодницької гри.

					ДП. КН 21.433.02.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Історія жанру. Бретт Вайс цитує Супермена Atari (1979) як екшен-пригодницьку гру, а Retro Gamer зараховує її як "першу, яка використовує кілька екранів як ігрову зону". Марк Дж. П. Вольф приписує Adventure (1980) для Atari VCS як найдавнішу відому екшен-пригодницьку гру. Гра передбачає вивчення 2D-середовища, пошук і використання предметів, котрі мають передбачені здібності, і боротьба з драконами в режимі реального часу, як у екшн-грі. Замок Wolfenstein від Muse Software (1981) - ще одна рання екшн-пригодницька гра, що поєднує елементи дослідження, бою, стелсу та лабіринтів черпаючи натхнення в аркадних стрілянинах та іграх у лабіринті (наприклад, стрілянина в лабіринті Берзерк) та військових фільмах (таких як Гармати Навароне).

За словами розробника Wizardry Po Р. Адамса, ранні екшен-пригодницькі ігри "в основному були аркадними іграми, виконаними у фантастичній обстановці". Тутанкхем, дебютований Конамі в січні 1982 р., був пригода випущена для аркад.

Він поєднував лабіринт, стрілянину з них, розгадування головоломки та елементи пригод з оглядом журналу Computer and Video Games 1983 року, називаючи його "першою грою, яка ефективно поєднала елементи пригодницької гри з несамовитою стріляниною вгору до геймплею ". Це надихнуло подібного Time Bandit (1983). Action Quest, випущений у травні 1982 року, поєднав елементи головоломки пригодницьких ігор у екшн-гру в аркадному стилі, керовану джойстиком, що на той час здивувало рецензентів.

Відзначаючи деяку схожість з Adventure, IGN стверджує, що The Legend of Zelda (1986) від Nintendo "допомогла створити новий піджанр екшн-пригод", досягнення успіху завдяки тому, що він поєднав елементи з різних жанрів, щоб створити переконливий гібрид, включаючи дослідження, головоломки в стилі пригод, компонент дії, грошову систему та спрощену побудову рівня RPG без очок досвіду. Серія «Легенда про Зельду» стала найпродуктивнішим бойовиком - франшиза пригодницьких ігор до 2000-х років. Роу Р. Адамс також наводив аркадні фантастичні ігри на боковій прокрутці Castlevania (1986), Trojan (1986) та Wizards & Warriors (1987) як ранні приклади екшен-пригодницьких ігор.

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Такі ігри, як Brain Breaker (1985), Xanadu (1985), Metroid (1986) та Vampire Killer (1986), поєднували в собі прокручування формату платформера з пошуком пригод, створюючи піджанр платформи-пригоди Metroidvania. Подібним чином, такі ігри, як 005 (1981), Castle Wolfenstein та Metal Gear (1987), поєднували екшн-пригодницькі дослідження з механікою стелс, закладаючи основи піджанру стелс-гри який згодом буде популяризований у 1998 році випусками Metal Gear Solid, Tenchu: Stealth Assassins та Thief: The Dark Project.

Кінематографічний платформер «Принц Персії» (1989) містив екшен-пригодницькі елементи, надихаючи такі ігри, як «Інший світ» (1991) та «Флешбек» (1992). Alone in the Dark (1992) використовував 3D-графіку, який згодом буде популяризований Resident Evil (1996) та Tomb Raider (1996). Resident Evil, зокрема, створив піджанр жахів виживання, надихаючи такі заголовки, як Silent Hill (1999) та Fatal Frame (2001). Екшн-пригодницькі ігри стали популярнішими, ніж чисті пригодницькі ігри та ігри на платформі, які їх надихнули.

Різновиди: графічна пригода, point & click adventure, головоломки, візуальні новели.

Графічна пригода: ці типи пригодницьких ігор були дуже популярні протягом 1980-х років. Я збираю їх сюди, бо; а) авантюри з чистим текстом не є частиною цього джему, тому їм не потрібен власний розділ; б) вони грають приблизно так само. Велика різниця між ними, звичайно, полягає в тому, що текстові пригоди описують те, що відбувається в тексті, тоді як графічні пригоди показують це на екрані, і дозволяють вам рухати персонажа безпосередньо, а не вводити вказівки типу "йти на північ" або "зайти у двері".

Ці ігри визначили, що ми зараз розуміємо під "пригодницькою грою". Вони інтенсивно орієнтовані на історію і вимагають від гравця розгадування головоломок, щоб просувати історію. Механіка включає збір предметів та їх складання в інвентар. Потім ці предмети використовуються у світі для досягнення цілей. Усі команди набираються як текст типу "отримати ключ" і "відкрити двері ключем".

					ДП. КН 21.433.02.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Оскільки команди вводяться користувачем як текст, і майже все може бути виражене мовою, типи головоломок та взаємодій, можливих у цих іграх, потенційно безмежні. Це робить геймплей набагато гнучкішим та варіативнішим, ніж пізніші пригоди, натискання та клацання. Однак це має суттєвий мінус, в тому, що розробники повинні передбачити все, що користувачі можуть розумно ввести в ситуації, включаючи всі синоніми, які вони можуть використовувати. Це призводить до того, що гравці, як правило, бачать багато "Я не розумію цього, будь ласка, переробіть" або щось подібне після введення команди.

Добре відомі графічні пригоди синтаксичного аналізу тексту включають оригінальні ігри King's Quest, Space Quest та Leisure Suit Larry. Сучасний приклад – майбутня гра Cascade Quest від Icefall Games .

Point & Click Adventure. У 90-х роках пригодницькі ігри перетворилися на інтерфейс point & click, і це все ще є стандартом для пригодницьких ігор сьогодні. Замість того, щоб вводити команди, гравець вибирає з кількох взаємодій (або предметів інвентарю) і просто клацає об'єктом, з яким бажає взаємодіяти. Через більш обмежений характер головоломок можна було б зробити так, більшість головоломок вказують і клацають, як правило, це питання збору потрібних предметних запасів, а потім використання їх у потрібних місцях.

Ранні пригоди вказівниками та клацаннями часто намагалися зробити гру більш складною, приховуючи предмети або предмети в сцені, що призводило до необхідності прискіпливо полювати на пікселі кожну сцену, щоб знайти те, що вам потрібно, або розробляли головоломки, які були абсолютно абсурдними, щоб зробити їх неочевидними. Однак новітні ігри визріли з цього і знайшли кращі способи тримати гру складною, не роблячи її нудною.

Пригоди, спрямовані на натискання та клацання, мають, як правило, ще більш складні та захоплюючі історії та більш цікаві розкриті персонажі.

Таким чином, діалогові дерева (іноді досить складні) часто зустрічаються у пригодах із натисканням і натисканням, що дозволяє NPC ожити. Добре розроблені пригоди вказівниками та клацаннями відомі тим, що тісно включають загадки в розповідь, а не просто є довільними перешкодами для прогресування.

					ДП. КН 21.433.02.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

До добре відомих пригод за допомогою точки та клацання можна віднести пізніші ігри серії King's Quest та Space Quest, Day of The Tentacle та Grim Fandango. Сучасні приклади численні і включають Paradigm, Bear With Me та будь-яку гру, опубліковану Wadjet Eye Games.

Головоломки. Головоломні пригодницькі ігри - єдиний тип, при якому історія займає більше заднього місця. Це все ще історія (інакше це не була б пригодницька гра!), Але головоломки самі попереду і в центрі. У пригодницькій грі-головоломці рішення головоломок значною мірою від'єднані від історії, за винятком якихось довільних способів. З цієї причини, історія часто включає антагоніста, який розробляв головоломки, які гравець тепер повинен розгадати. Насправді справжнім антагоністом головоломок є загадки.

Головоломки в цьому типі гри, як правило, являють собою серію окремих логічних головоломок і, як правило, набагато складніше, ніж головоломки в інших типах пригодницьких ігор.

Цей тип гри рідко має NPC, принаймні того типу, з яким ви можете взаємодіяти.

Добре відомі головоломні пригоди включають Myst та The 7th Guest. Інтерактивний візуальний роман (або гра "вибери свою пригоду"). Ці ігри є протилежністю головоломці.

Вони містять мало загадок або взагалі не містять їх, а натомість повністю зосереджуються на історії та виборах, які робить гравець. Більшість цих ігор орієнтована на діалог, причому переважна більшість варіантів вибору гравцем робиться тим, що сказати або як реагувати на іншого персонажа, і, як правило, кілька важливих рішень у ключових моментах гри.

Більше, ніж у попередніх типів, в цих іграх вкрай важливо, щоб вибір гравців мав реальний результат для самої історії, часто включаючи не лише кілька закінчень, але і безліч різних шляхів через історію, залежно від вибору гравця, і, отже, гравцеві повинні бути запропоновані значущі варіанти протягом гри які глибоко впливають на історію.

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Добре відомими інтерактивними візуальними романами є «Життя дивне» та будь-яка гра Telltale, створена з 2012 року (до 2012 року Telltale робила точки і клацання).

Візуальні новели. Я особисто вважаю, що це більше виділення жанру пригодницької гри, але вони чітко пов'язані та прийнятні для цього джему.

Визначальною характеристикою цього типу є те, що гравець не може багато чого зробити, крім як ходити і розглядати речі. Як правило, не існує NPS, з якими можна взаємодіяти, або взагалі немає головоломок, або кілька тривіально простих головоломок, і немає значущого вибору гравцем. Ці ігри стосуються пасивного переживання навколишнього середовища та історії.

Часто це тому, що історія траплялася в минулому, і персонаж гравця не був її частиною. Гравець просто знаходить підказки про те, якою була історія, коли вони просувалися по світу.

## 1.2 Аналіз наявних рішень

Пригодницька гра – це відеогра, в якій гравець бере на себе роль головного героя в інтерактивній історії, що рухається дослідженням та вирішенням головоломок.

Зосередженість жанру на сюжеті дозволяє йому сильно спиратися на інші засновані на розповіді засоби масової інформації, літературу та кіно, охоплюючи широкий спектр літературних жанрів:

- графічні квести,
- пригодницькі ігри вказуй і клацни,
- втеча з кімнатних ігор,
- головоломки пригодницькі ігри,
- розповідні пригодницькі ігри,
- симулятор бігу.

Багато пригодницьких ігор (текстових та графічних) розроблені для одного гравця, оскільки такий акцент на сюжеті та персонажах ускладнює

дизайн для багатьох гравців. Colossal Cave Adventure визначена як перша така пригодницька гра, вперше випущена в 1976 році, тоді як інші помітні серії пригод включають Zork, King's Quest, Секрет острова мавп і міст. На рисунку 1.1 та 1.2 зображені логотипи цих ігор.

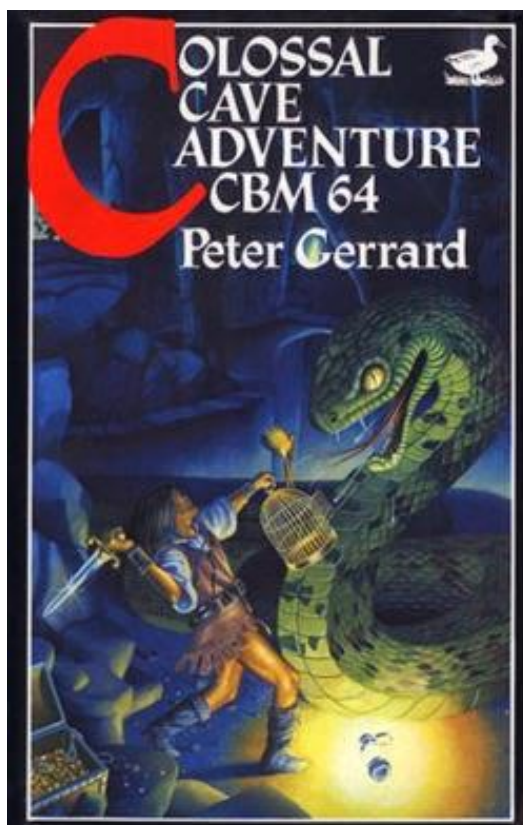


Рисунок 1.1 – Логотип гри «Colossal Cave Adventure»



Рисунок 1.2 – Логотип гри «Секрет острова мавп і міст»

					ДП. КН 21.433.02.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

На рисунку 1.3 зображена гра «80 днів».

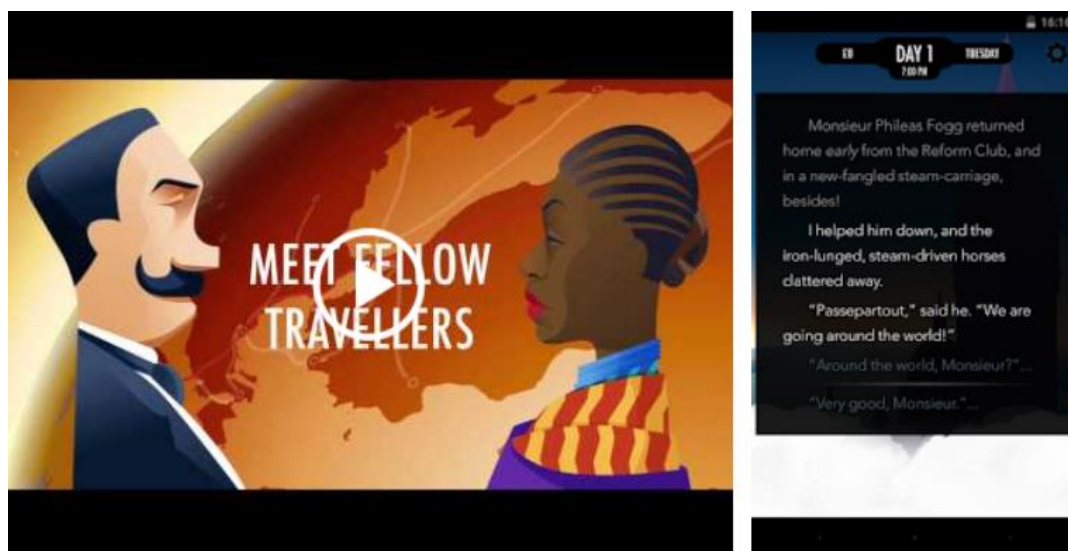


Рисунок 1.3 – Логотип гри «80 днів»

80 днів – одна з найкращих пригодницьких ігор для Android. Ви граєте в ролі помічника Філеаса Фогга, Паспарту, і намагаєтесь обійти земну кулю всього за 80 днів. Ви можете сісти на дирижаблі, підводні човни, поїзди і навіть механічні верблюди.

Гравці повинні керувати життєвими показниками Філеаса, коли ви подорожуєте по всьому світу або ризикуєте програти гру. Гра грається швидкими темпами, і ви буквально шукаєте пригод по всьому світу. Крім того, ви також змагаєтесь з іншими гравцями, щоб побачити, хто там першим потрапить. Це єдиний у своєму роді тип гри, який вартий свого цінника.

На рисунку 1.4 зображено гру Крахи.



Рисунок 1.4 – Логотип гри «Крахи»

					ДП. КН 21.433.02.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		



Crashland вже є однією з найкращих пригодницьких ігор 2016 року. Ви граєте як космічний далекобійник на ім'я Flux. Гра починається з того, що ви вриваєтесь на чужу планету. Ваша робота - отримати свій вантаж, побудувати собі базу, боротися з поганими хлопцями, і врятувати світ. У ньому багато ігрових елементів, включаючи крафт, бойові дії, вирівнювання персонажів у стилі RPG, і ви навіть можете приручити істот, щоб битися за вашу сторону. Це глибока гра з великою кількістю вмісту, яку ви можете грати як на мобільному, так і на ПК. Рейтинг Google у Google Play - 4,8.

На рисунку 1.5 зображено гру Evoland 2.

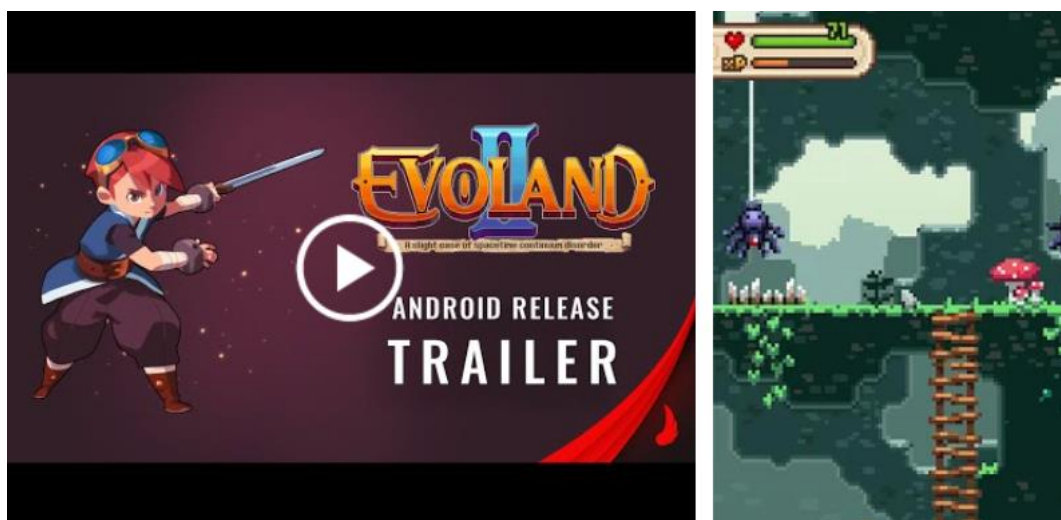


Рисунок 1.5 – Логотип гри «Evoland 2»

Evoland 2 легко є однією з найкращих пригодницьких ігор за останній час. Він виходить за межі жанрів, включаючи цілу купу з них. Сюди входять 2D RPG, 3D-винищувач, торгова картка, рубати і багато інших.

Гра в основному прогресує через різні епохи ігор, і кожна нова механіка також привносить новий графічний стиль. Унікальна гра дозволяє змусити вас здогадуватися, і це приємний загальний досвід.

Однак для найкращих результатів ми рекомендуємо грати з фізичним контролером.

					ДП. КН 21.433.02.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

На рисунку 1.6 зображено гру Вплив Геншина.



Рисунок 1.6 – Логотип гри «Вплив Геншина»

Genshin Impact - одна з нових пригодницьких ігор у списку. Це дуже схоже на Zelda: Breath of the Wild. Гравці отримують великий, відкритий світ для вивчення за допомогою безлічі квестів та місій. Гравці навіть отримують планер для подолання більших відстаней. У цьому також є елемент RPG gacha, але ви отримуєте більшість персонажів за допомогою квестів, а не випадкових викликів, тому це додає трохи смаку в гру. Це виглядає добре, управління відмінне, і ви будете подорожувати цим довгий час.

На рисунку 1.7 зображено гру Велика гірська пригода.

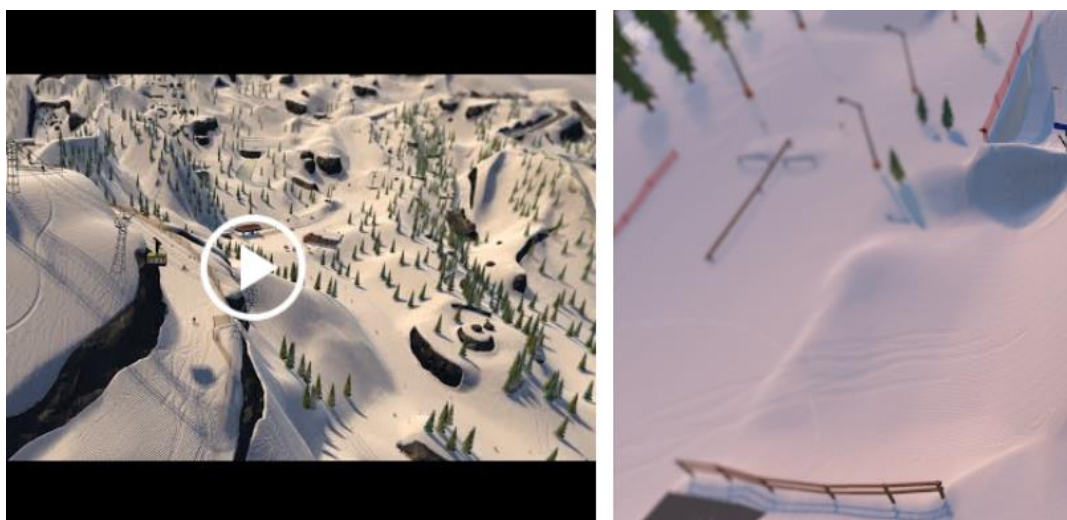


Рисунок 1.7 – Логотип гри «Велика гірська пригода»

					ДП. КН 21.433.02.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Grand Mountain Adventure - одна з нових пригодницьких ігор у списку. Це лижна гра з гоночними та дослідницькими елементами. Гравці катаються на ліфті до різних трас.

Потім ви катаєтесь на лижах на трасах і намагаєтесь перевершити час. Однак ви також можете просто кататися на лижах і розглядати визначні пам'ятки. Також є приховані ділянки, п'ять гір (у преміум-версії), онлайн-таблиці лідерів, і ви навіть можете робити трюки, коли потрапляєте у велике повітря. Управління і перспектива гри спочатку трохи дивні, але не надто складні. Графіка також чудова, а гра - плавна. Ви отримуєте першу гору безкоштовно, і ви можете придбати преміум-версію, щоб розблокувати решту.

### 1.3 Постановка задачі

Більше ніж будь-який інший жанр, пригодницькі ігри залежать від їх історії та обстановки, щоб створити переконливий досвід для одного гравця. Вони, як правило, розміщуються в захоплюючому середовищі, часто у світі фантазій, і намагаються змінювати обстановку від глави до глави, щоб додати новизни та інтересу до досвіду.

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ ПРОДУКЦІЇ

### 2.1 Формалізація вимог до системи

Вимоги до системи визначають умови та можливості, яким повинна відповідати розроблена система. Вони є частковим визначенням у природній мові з неминучими двозначностями. Формалізація стосується перетворення вимог у специфікацію з унікальною інтерпретацією для вирішення неясностей, недостатньо визначені посилання та для оцінки того, чи вимоги є послідовними, правильними (тобто дійсними для прийнятного рішення) та досяжними. Формалізація та валідація системних вимог дає ранні докази адекватних специфікацій для зменшення тестів валідації та високих коригувальні заходи на пізніх етапах розробки системи. Ця стаття містить наступні матеріали. По-перше, ми характеризуємо проблему специфікації на основі онтології для деякої області. Таким чином, вимоги представляють певну систему серед багатьох можливих, а їх специфікація набуває форми відображення їх концепцій у семантичній моделі системи. По-друге, ми аналізуємо найсучасніші мови специфікацій на основі шаблонів, які використовуються для уникнення двозначності. Потім ми обговорюємо семантичний аналіз (відсутні вимоги, невідповідності тощо), підтриманий у такій структурі. По-третє, ми проводимо опитування пов'язаних досліджень щодо виведення формальних властивостей із вимог, тобто перевірених специфікацій, які обмежують структуру та поведінку системи. Можливі недоліки у вимогах можуть зробити похідні властивості незадовільними або нереалізованими. Нарешті, у цій статті розглядаються важливі виклики сучасним інструментам аналізу вимог, які мають бути прийняті в промислових масштабах.

Інженерія вимог має життєво важливе значення в системній інженерії. Він складається з виявлення, уточнення та управління вимогами з метою створення системи, яка відповідає потребам зацікавлених сторін.

Системні вимоги визначають, як система відповідатиме вищим рівень вимог до зацікавлених сторін (їх також називають ранніми вимогами). Ці вимоги є

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

частковою специфікацією системного рішення проєктної проблеми, яка не обмежується конкретною конструкцією. Оскільки вимоги зазвичай пишуться контрольованою природною мовою (тобто обмежені синтаксисом або лексичними термінами), вони неминуче неоднозначні. Це підриває нашу здатність гарантувати основні якості, такі як відсутність невизначених посилань на системні сутності та невідповідності, а також те, що специфікація дійсна для прийнятного рішення (правильного) та досяжного.

Можливі недоліки системних вимог передбачають повторення дорогих циклів перевірки валідації та виправлення конструкції системи на пізніх етапах розробки системи.

Тому потенціал належної перевірки та вдосконалення системних вимог на початку циклу розробки та застосування цього до промислово-масштабні проєкти – це виклик, який ще не був належним чином розглянутий за сучасного стану практики. Валідація пов'язана з проблемою перетворення вимог у формальну специфікацію, що підлягає верифікації (формалізація вимог).

Це означає, що всі виявлені неясності та невизначені посилання були вирішені, тоді як було визначено відображення в точно визначену семантичну модель системи. Тому ми припускаємо, що специфікація відбувається з посиланням на модель домену, тобто онтологія з точно визначеними логічними взаємозв'язками та фактами щодо понять, згаданих у вимогах домену системи. Потенційні двозначності можна усунути, якщо використовувати мову шаблонів вимог із чітко визначеною семантикою. Потім можуть бути підтримані різні семантичні аналізи, такі як виявлення випадків відсутності вимог, перевірка відсутності суперечливих специфікацій, виявлення термінів, які слід замінити більш конкретними підкласами концепції тощо.

Однак, навіть після застосування цих аналізів ще не існує жодної гарантії відсутності специфічних недоліків; хоча вони покращують системні вимоги на семантичному рівні, вони не можуть зробити їх перевіреними. Це може статися лише в тому випадку, якщо системні вимоги можна задовольнити та реалізувати, що може бути вирішено після того, як вони будуть перетворені у формальну

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

специфікацію. Це відбувається під час збору системних вимог за властивостями мовою логіки, для очікуваної поведінки та структури правильної конструкції системи.

Хоча вимоги повинні бути незалежними від конкретної конструкції системи, властивості виражаються у формальній формі (тобто абстрактне подання мовою специфікації з формальною семантикою) системи, що розробляється, і фактично обмежує дизайн системи. Виведення властивостей може базуватися на мові шаблонів властивостей, де кожному шаблону присвоєна семантика логічною мовою.

Кожна окрема вимога охоплюється властивостями, які можуть бути отримані за допомогою шаблонів властивостей, пов'язаних із конкретними шаблонами специфікації вимоги, та шляхом асоціювання концепцій вимоги з подіями формальної моделі системи. Це гарантує, що всі вимоги мають послідовне тлумачення щодо формальної моделі системи. Якщо модель не задовольняє властивостей, тоді слід дотримуватися модифікованої конструкції (та моделі) або уточнювати певні вимоги, які не задовольняються.

Ця стаття висвітлює всі згадані проблеми формалізації та перевірки системних вимог, оглядає найсучасніші сучасні мови та обговорює існуючі мови шаблонів, доступні інструменти та їх обмеження щодо можливості їх застосування в промислових проєктах.

Решта статті структурована таким чином. У наступному розділі розглядаються проблеми двозначності та недостатньої конкретизації вимог до природної мови. У розділі представлені останні досягнення в онтологічному моделюванні системних вимог, їх специфікація за допомогою відповідних мов шаблонів та підтримуваний семантичний аналіз.

Ця проблема пов'язана з проблемою проєктування системи, і з цієї причини ми також представляємо дві основні парадигми модельної та компонентної на основі проєктування системи.

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2.2 Проектування структури систем

Проектування та структура ігрового додатку починається з вибору програмного забезпечення, на якому буде створено. На теперішній час представлений дуже великий вибір програмних забезпечень та крос-платформ для створення ігрового мобільного додатку. Щоб створити мобільну гру було обрано таку платформу як Corona Simulator. В першу чергу розпочати потрібно з того, що саму являє собою Corona Simulator.

Легкий у вивченні та потужний Corona - це платформа, що ідеально підходить для швидкого створення програм та ігор для мобільних пристроїв та настільних систем.

Це означає, що ви можете створити свій проєкт один раз і опублікувати його на різних типах пристроїв, включаючи Apple iPhone і iPad, телефони та планшети Android, Amazon Fire, Mac Desktop, Windows Desktop і навіть підключені телевізори, такі як Apple TV, Fire TV та Android TV.

Corona – це безкоштовний кроссплатформенний фреймворк, ідеально підходить для створення ігор та програм для мобільних пристроїв та настільних систем.

Використовуючи потужну, але просту у вивченні мову сценаріїв Lua, понад 1000 вбудованих API, величезний вибір плагінів та розширення Corona Native (C / C ++ / Obj-C / Java), можна втілити мрії своєї програми в реальність. В ній навіть можна миттєво відстежувати проєкти на декількох пристроях за допомогою Live Builds.

Corona використовує Lua, потужну та просту у вивченні мову сценаріїв, підкріплену великою документацією. Lua розроблений як легкий, швидкий, але при цьому потужний. Він використовується у таких головних назвах, як Civilization, Warcraft та незліченних незалежних іграх. Lua є дуже доступна і пропонує просто, прощення синтаксису – якщо ви ніколи раніше не кодували, Lua та Corona – ідеальне місце для початку!

На рисунку 2.1 зображено функціональну діаграму проєкту.

					ДП. КН 21.433.02.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

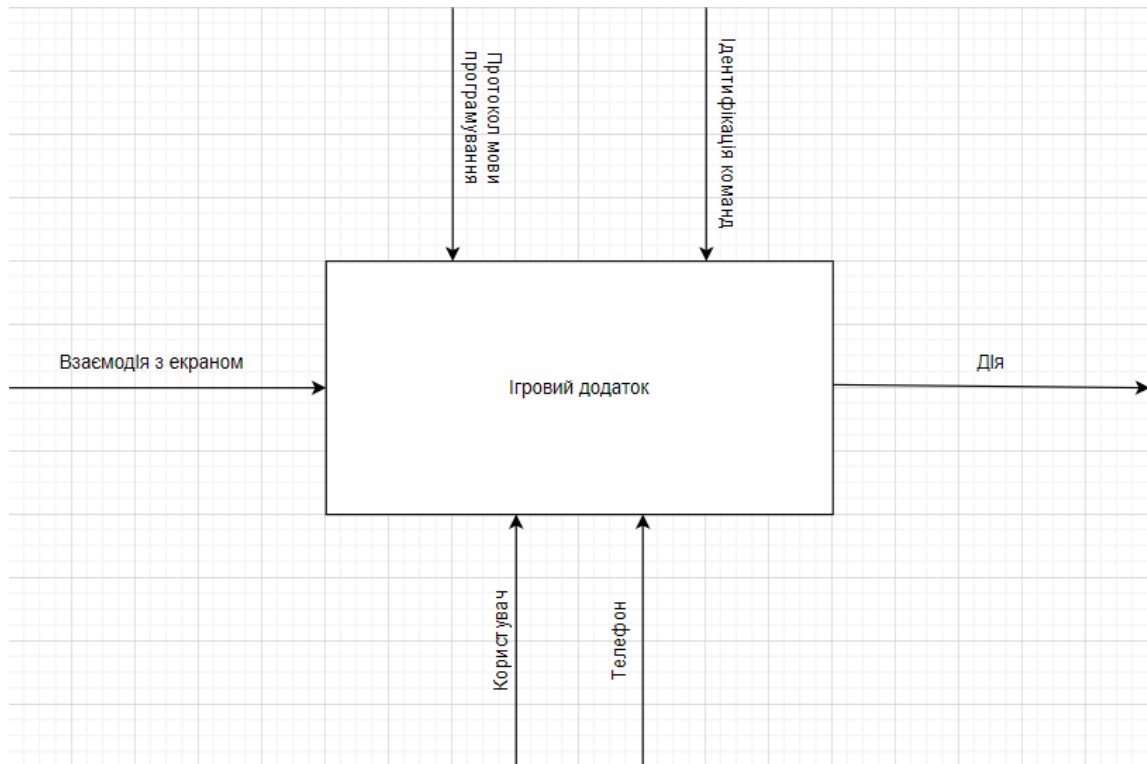


Рисунок 2.1 – IDEF0 діаграма проєкту

### 2.3 Проєктування алгоритму роботи систем

Алгоритм роботи системи – це кількість інструкцій, які відображають послідовність певних дій користувача, щоб розв’язати ту чи іншу задачу за певну кількість дій; система правил дискретного процесу, яка добивається мети за скінченну кількість часу.

Для початку поговоримо про механіку алгоритмів, яка є не складною, всього лиш потрібно взаємодіяти з екраном смартфона, після чого здійснюється анімація 2D-персонажа на якій він рухається вперед.

Під час руху в рандомному порядку будуть з’являтися перешкоди взаємодіючи з якими гравець закінчує свою гру і отримує певну кількість очків, а саме певну кількість пройдених метрів.

На рисунку 2.2 та 2.3 зображені діаграми руху 2D-персонажа та генерації ігрової локації.



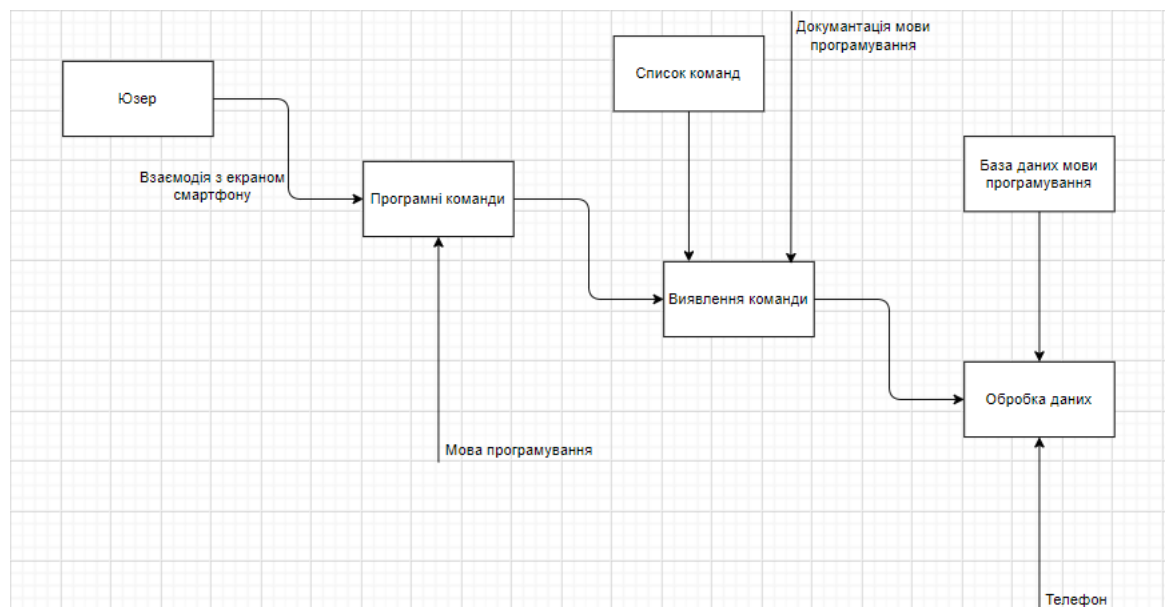


Рисунок 2.2 – DFD діаграма руху ігрового 2D-персонажа

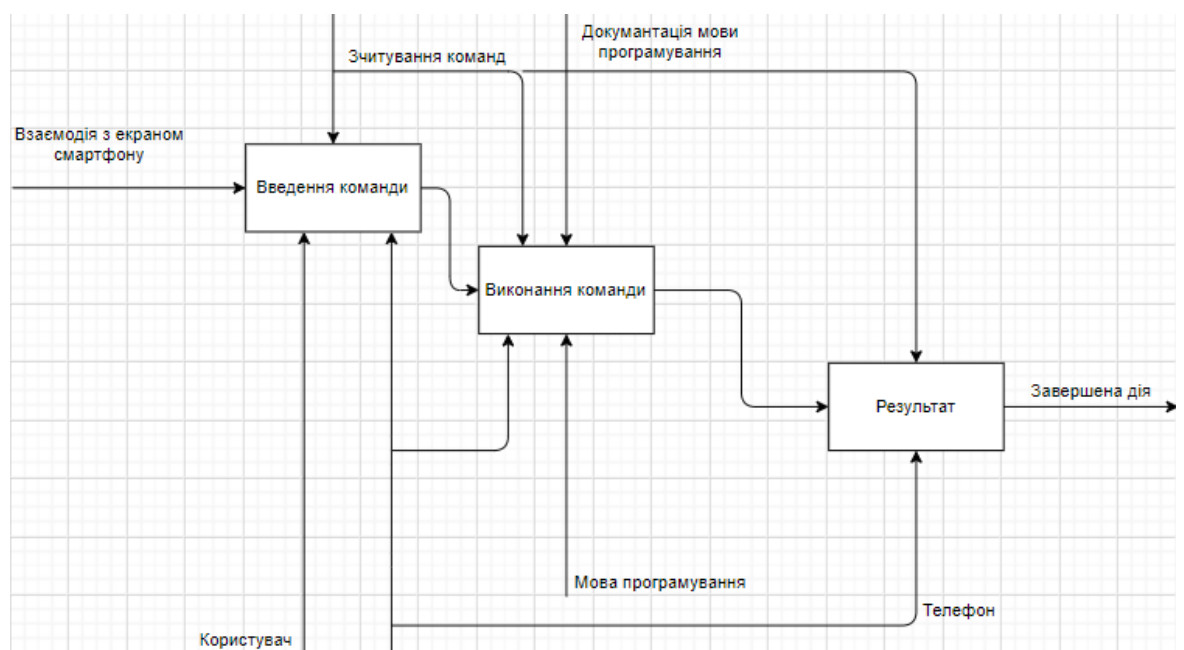


Рисунок 2.3 – DFD діаграма генерації ігрової локації

### 3 РЕАЛІЗАЦІЯ СТВОРЕНОЇ СИСТЕМИ

#### 3.1 Аналіз інструментальних засобів для реалізації гри

У цій роботі поставлено завдання розглянути сучасні інструментальні засоби для розробки ігор. Було проаналізовано найпопулярніші сучасні ігрові рушії, фреймворки та конструктори ігор.

Планшети та смартфони вже стали звичними речами, як раніше домашні ПК. В результаті вони відкривають дивовижні поля для ігрової індустрії. Тут інструменти розробки мобільних ігор можуть стати в нагоді для реалізації ваших ідей.

Поки технології розвиваються щодня, усі ці мобільні пристрої стають все більш досконалішими та зручнішими у використанні. Останнім часом ринок настільних ігор почав втрачати свої позиції через зростання ринку мобільних ігор. В результаті ринок розробки мобільних ігор щодня розширюється та наповнюється новими продуктами.

Однак конкуренція на ринку також зростає. Отже, з середнім продуктом помітити важче.

Щоб досягти успіху, бажана гра повинна бути більшою за будь-які засоби. Але сьогодні двигун і графіка такі ж важливі, як і геймплей, і сюжет. Отже, для отримання вам потрібні відповідні інструменти розробки для вашої гри.

Ми поділили всі інструменти, придатні для створення ігор, на кілька категорій. Ось їх список із кількома прикладами для кожного з них.

#### 3.2 Вибір засобів реалізації

Ігровий дизайн є однією з ключових сфер індустрії цифрових розваг. З появою нових технологій виробничі інструменти значно еволюціонували, але мало що зроблено для підтримки ігрового дизайну. Дизайнери все ще використовують ті самі інструменти з початку району, і хоча промисловість зазнала певних змін послідовні успіхи в продажах, дослідники та професіонали сходяться на думці, що відсутність інструментів, будь то концептуальних чи

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

програмних, створює перешкоду для будь-якої спроби стандартизації та перешкоджає передачі знань між поколіннями дизайнерів. Вони запропонували концептуальні та конкретні інструменти, які могли б доповнити або замінити проєктний документ, з метою вдосконалення процесу ігрового створення. У цій роботі представлена систематизація цих напрямків через хронологічний огляд основних підходів та їх реалізації для того, щоб визначити їх у контексті сучасного рівня дизайнерських інструментів.

На даний момент існує безліч цифрових та програмних забезпечень для створення ігрового продукту. Нижче перелічені декілька з цих забезпечень.

Game Maker – серія крос-платформених ігрових двигунів (рисунок 3.1).



Рисунок 3.1 – Логотип Game Maker

GameMaker (спочатку Animo, а пізніше Game Maker до 2011 року) – це серія крос-платформених ігрових движків, створена Mark Overmars в 1999 році і розроблена YoYo Games з 2007 року. Остання версія - GameMaker Studio 2, вперше випущена в 2017 році.

GameMaker дозволяє створювати крос-платформні та багатожанрові відеоігри за допомогою спеціального перетягування і відпустіть мову візуального програмування або мову сценаріїв, відому як Game Maker Language, яка може

бути використана для розробки більш досконалих ігор, які неможливо створити лише за допомогою функцій перетягування.

GameMaker спочатку був розроблений, щоб дозволити початківцям комп'ютерним програмістам створювати комп'ютерні ігри без особливих знань програмування, використовуючи ці дії. Останні версії програмного забезпечення також зосереджені на зверненні до просунутих розробників.

Clickteam Fusion – компанія з виробництва програмного забезпечення (рисунок 3.2).



Рисунок 3.2 – Логотип Clickteam Fusion

Clickteam - французька компанія з виробництва програмного забезпечення, заснована в 1993 році Франсуа Ліоне (фр. François Lionet), Івом Ламуре (фр. Yves Lamoureux) і Франсісом Пуленом (фр. Francis Poulain) зі штаб-квартирою в Булонь-Біянкур, Франція.

Clickteam відома за створення програмного забезпечення Clickteam Fusion — грального рушія та редактора, у якому можна створювати двовимірні відеоігри без використання скриптів.

					ДП. КН 21.433.02.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

Construct 2 – 2D-редактор ігор (рисунок 3.3).



Рисунок 3.3 – Логотип Construct 2

Construct - це 2D-редактор ігор на основі HTML5, розроблений Scirra Ltd. Він орієнтований насамперед на непрограмістів, що дозволяє швидко створювати ігри за допомогою візуального програмування. Вперше випущений як ліцензований GPL ігровий движок DirectX 9 для Microsoft Windows з програмуванням на Python 27 жовтня 2007 р. пізніше воно стало власним програмним забезпеченням з Construct 2, а також переключило свою технологію API з DirectX на NW.js та HTML5, а також видалило Python та додало підтримку JavaScript та його плагін SDK у 2012 році, і врешті-решт перейшло на підписку модель на базі веб-програми (рисунок 3.3).

					ДП. КН 21.433.02.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

Solar2D – набір крос-платформних програм для розробки програмного забезпечення (рисунок 3.4).



Рисунок 3.4 – Логотип Solar 2D

Solar2D (раніше Corona SDK) - це безкоштовний набір крос-платформних програм для розробки програмного забезпечення з відкритим кодом, спочатку розроблений Corona Labs Inc. Випущений наприкінці 2009 року, він дозволяє програмістам створювати двовимірні мобільні додатки для iOS, Android та Kindle, настільні програми для Windows, Linux та macOS, та підключені телевізійні програми для Apple TV, Fire TV та Android TV (рисунок 3.4).

Solar2D використовує інтегровану Lua, шарувату поверх C++ / OpenGL, для створення графічних додатків. Програмне забезпечення має два режими роботи: Solar2D Simulator та Solar2D Native. За допомогою Solar2D Simulator додатки створюються безпосередньо з Solar2D Simulator.

Solar2D Native дозволяє інтегрувати код Lua та ресурси в проєкт Xcode або Android Studio для створення програми та включення власних функцій.

### 3.3 Реалізація програмного забезпечення

Для того, щоб швидко швидко та якісно просуватись в розробці проєкту потрібно створити план, який складається з таких пунктів, як: опис ігрового процесу, елементи управління ігрового додатку, анімація та звуки.

Створення гри буде відбуватись в такій програмі як Solar 2D. Починається все зі створення нового проєкту під назвою «DANGER Заєць». На рисунку 3.5 зображена панель налаштування нового проєкту в Solar 2D.

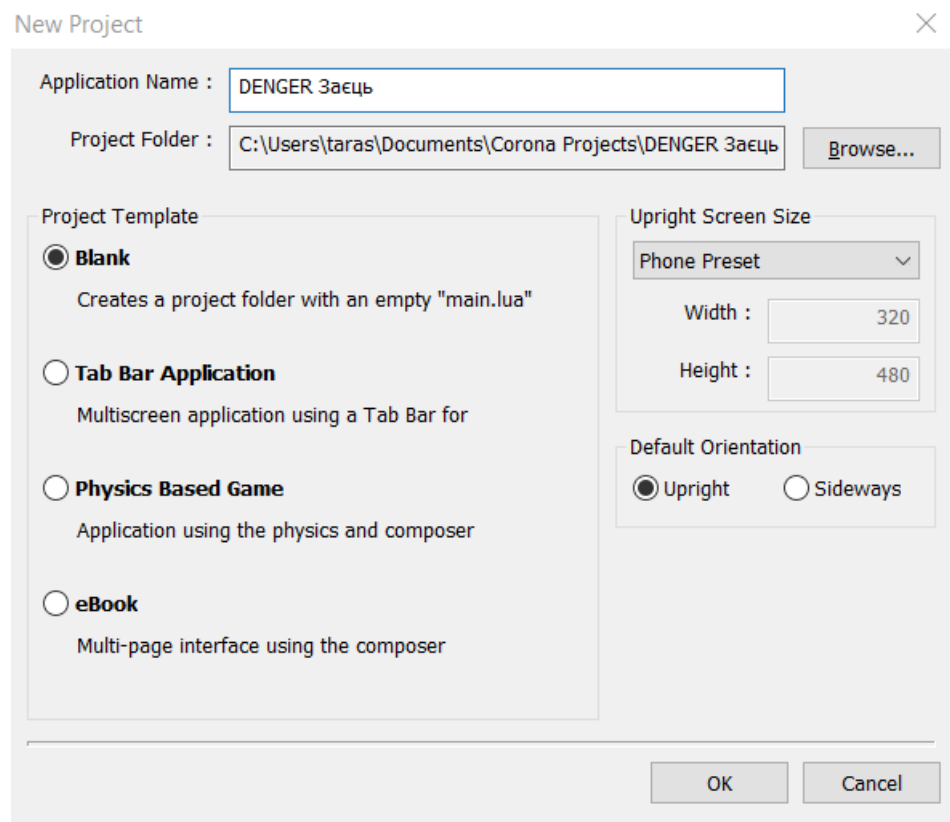


Рисунок 3.5 - Панель налаштування в Solar 2D.

Наступним кроком є додавання текстур до самого проєкту. Всі текстури йдуть з розширенням .PNG, які зображені на рисунку 3.6, деякі з них йдуть одним файлом щоб зменшити розмір гри та затрати енергії телефону.



Рисунок 3.6 – Об'єкти проєкту

					ДП. КН 21.433.02.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.4 Вибір способу захисту продукту

Захист ігрового додатку був реалізований через програму Java KeyStore.

Java KeyStore (JKS) - це сховище сертифікатів безпеки відкритих ключів, або сертифікатів авторизації, також відповідні приватні ключі, що використовуються, наприклад, у шифруванні TLS.

Java Development Kit підтримує файл сховища ключів CA з назвою cacerts у папці jre / lib / security. JDK надають інструмент з іменем keytool для управління сховищем ключів. keytool не має функцій для вилучення приватного ключа з магазину ключів, але це можливо за допомогою сторонніх інструментів, таких як jksExportKey, CERTivity, Portecle та KeyStore Explorer.

Для створення нового ключа потрібно запустити KeyStore Explorer і вибрати необхідний тип самого ключа, що зображено на рисунку 3.7.

На рисунках нижче (3.8, 3.9) відбувається саме створення нового ключа та його алгоритму.

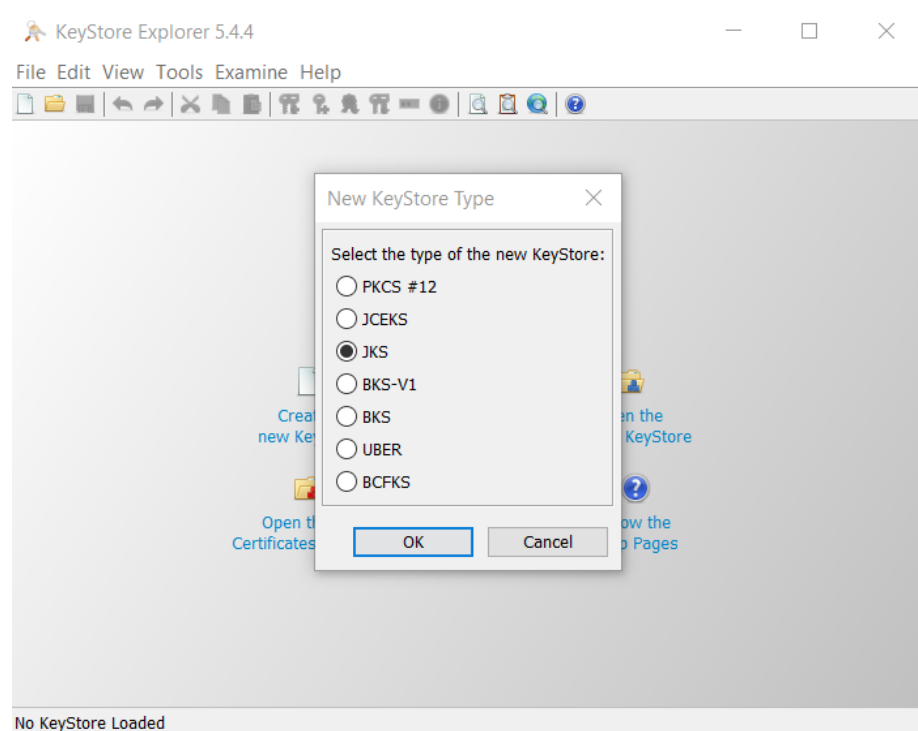


Рисунок 3.7 – Створення ключа



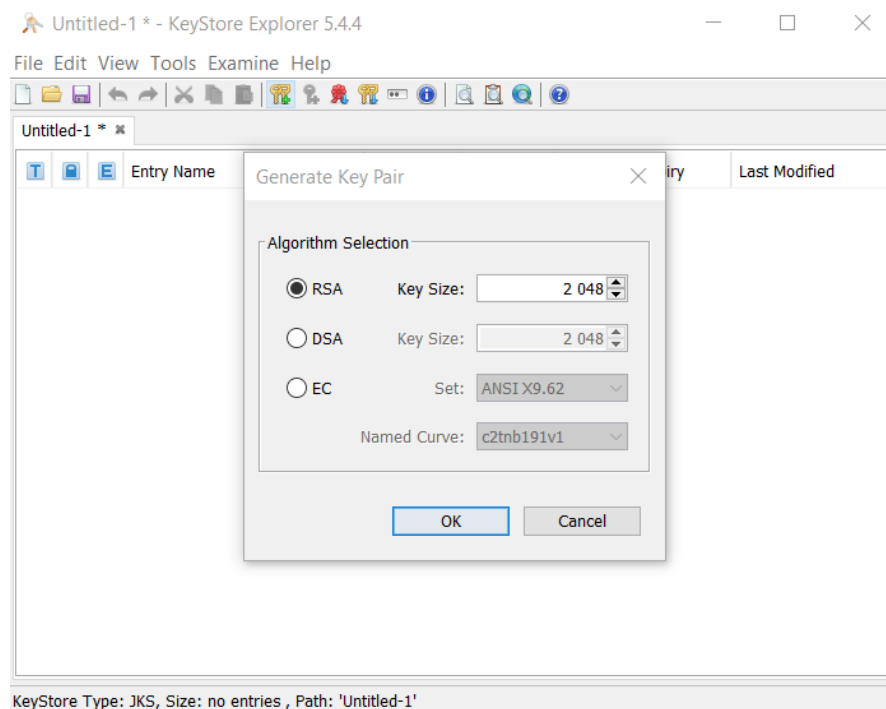


Рисунок 3.8 – Вибір нових ключів

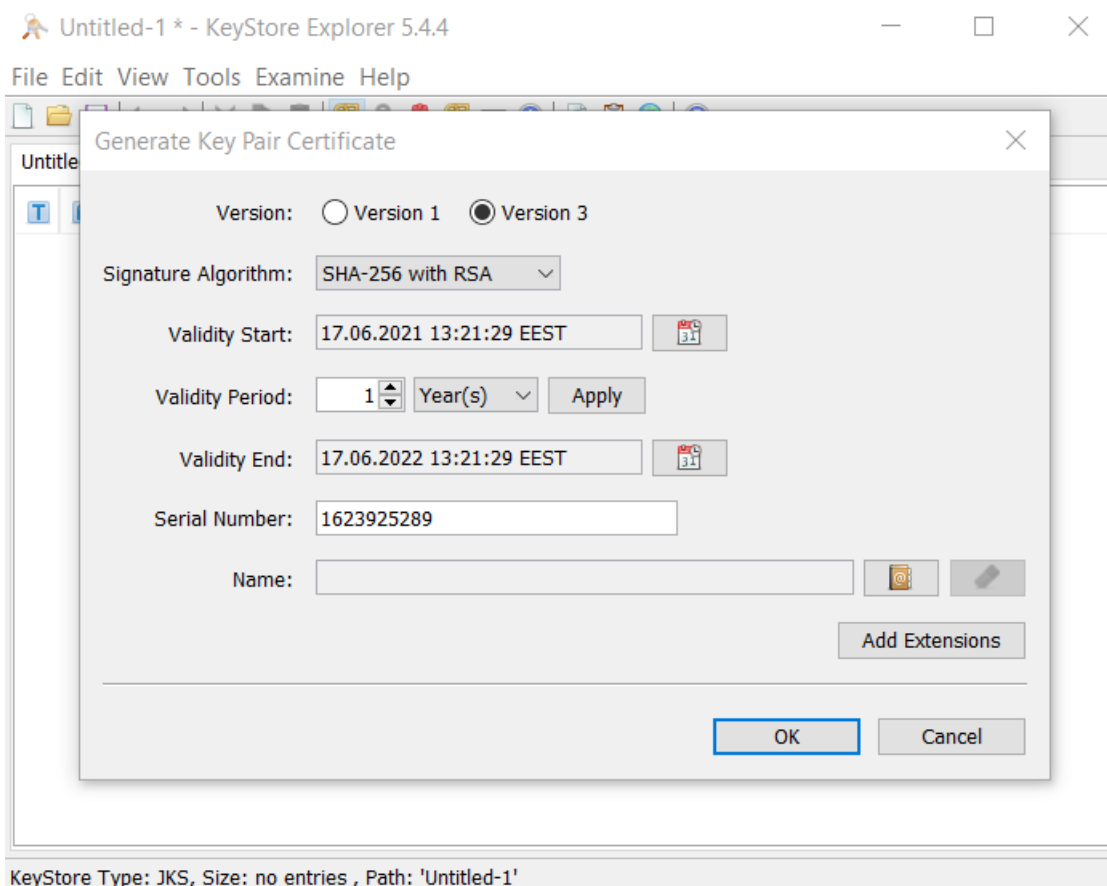


Рисунок 3.9 – Алгоритм ключа

Наступним кроком йде заповнення меню інформацією про свою організацію(компанію) та створення другого імені, що зображено на рисунку 3.10.

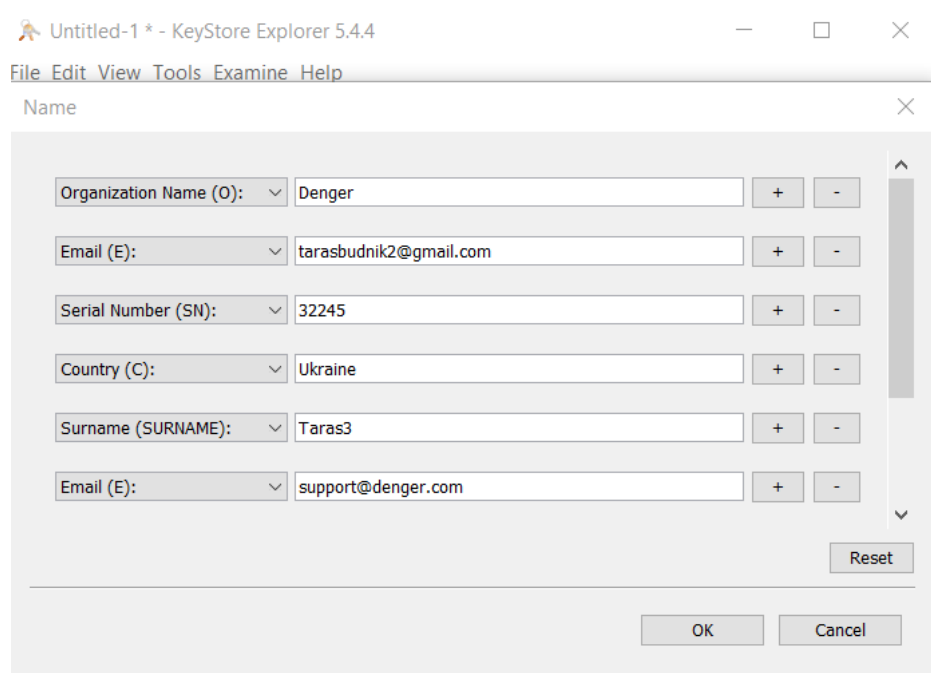


Рисунок 3.10 – Інформація про організацію

Після закінчення даних процедур збережений новий ключ можна знайти та ідентифікувати за другим ім'ям у сховищі.

Цей ключ можна використовувати і в наступних проєктах. Рисунок 3.11 відображає створений ключ.

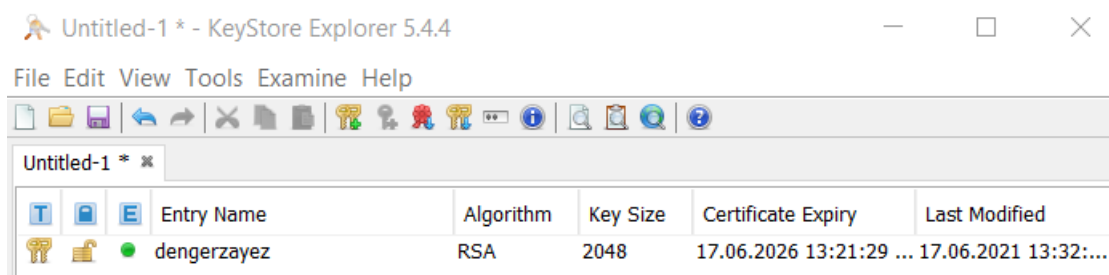


Рисунок 3.11 – Створений ключ

### 3.5 Тестування роботи системи

Опісля завершення над додаванням текстур об'єктів, функціоналом, захистом та вмістом інформації в грі відбувається тестування. Solar 2D – програмний емулятор, через який запускається даний проєкт.

На рисунку 3.12 зображене головне меню гри.

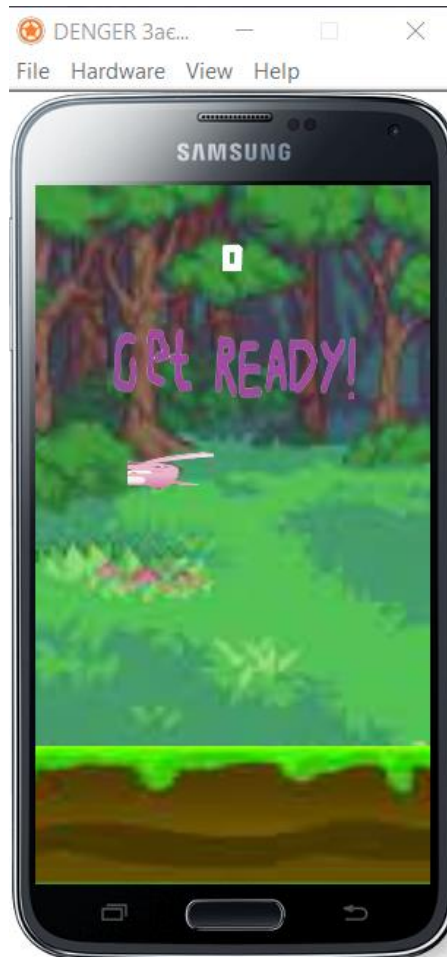


Рисунок 3.12 – Головне меню гри

Щоб почати гру достатньо натиснути на любую точку екрану смартфона. Після старту гравець повинен тиснути на екран смартфона та обминати перешкоди попереду.

Рисунок 3.13 відображає запуск гри.

					ДП. КН 21.433.02.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.13 – Проходження гри

Під час руху та обминання перешкод гравець отримує певну кількість очок в розмірі п'яти пройдених метрів. Це зображено на рисунку 3.14.

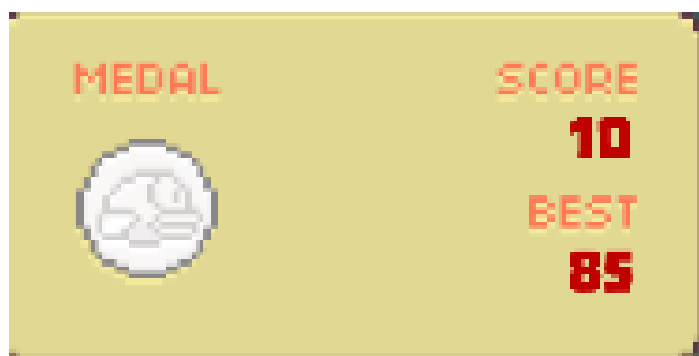


Рисунок 3.14 – Скрін набраної кількості очок

Підведемо підсумки. У процесі тестування гри не було виявлено жодного багу, так як робота самої гри була досить стабільною і без просадок. Анімація 2D-персонажа та самої корти під час руху була плавною. Тобто, в цілому можна сказати що проєкт вийшов досить успішним.

					ДП. КН 21.433.02.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 4.1 Аналіз ринку збуту продукту чи послуги.

За останній рік зростання кількості користувачів ігор приніс колосальні зміни в ігровій індустрії. Очевидним наслідком цієї зміни є те, що галузь отримує більше мільярдів доходів. Отже, у цього простору є величезні можливості для операторів, таких як розробники ігор, маркетологи тощо.

Якщо ви хочете скористатися цим, це дасть вам світ корисних для того, щоб бути поінформованими про відповідні тенденції та статистику. Маючи їх у своєму розумінні, поряд із знаннями галузі, ви зможете приймати розумні рішення та катапультиувати свій бізнес до успіху.

Епідемія сприяла зростанню індустрії мобільних ігор у 2020 році 2020 рік назавжди буде називатися роком пандемії.

Незважаючи на те, що бурхлива пандемія знищила багато підприємств у декількох галузях промисловості, ситуація, пов'язана з ігровою індустрією, була зворотною. Одним з найвідоміших шляхів, які люди обрали, є мобільні ігри.

Причини включають мобільні ігри, які є простішими та зручнішими за інші канали, забезпечуючи стільки ж задоволення.

#### Статистика мобільних ігор за 2020 рік

Щоб продемонструвати, наскільки швидко зросла індустрія мобільних ігор у 2020 році, ми розглянемо її статистичні дані. Як ви побачите, багато прибутків сталося.

Ми згрупували цю статистику за різними категоріями, щоб полегшити вам роботу.

#### Статистика доходів – 77,2 млрд. Доларів США доходу на 2020 рік

Індустрія мобільних ігор є однією з найефективніших галузей у світі в 2020 році. З масовою явкою 77,2 мільярда доларів у щойно завершився рік, немає жодних ознак уповільнення темпів у 2021 році.

#### Збільшення кількості гравців на 12%

					ДП. КН 21.433.02.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Індустрія мобільних ігор зафіксувала на 12% більше гравців у 2020 році, ніж у 2019 році, з понад 2,5 мільярдами гравців. Зважаючи на те, що пересування все ще обмежене, а люди обмежені домітками, передбачається, що цього року їх кількість зросте лише.

19 мільярдів доларів було витрачено лише у другому кварталі 2020 року.

Блокування спровокувало зростання кількості геймерів протягом другого кварталу попереднього року. У другому кварталі 2020 року геймери витратили на мобільні ігри понад 19 мільярдів доларів.

25% завантажень iOS та 21% програм Android.

Мобільні ігри становлять близько 21% усіх завантажень додатків для Android та 25% усіх завантажень додатків для iOS.

Оскільки багато інвесторів та розробників виходять із видатними іграми в 2021 році, цей відсоток, як очікується, зросте в 2021 році.

Тенденції покупки мобільних ігрових медіа.

2020 рік був вдалим роком для придбання мобільних ігрових ЗМІ, оскільки він пережив значне зростання. Наприклад, рекламодавці мобільних ігор зросли на 9% порівняно з попереднім роком до 60 000.

Ці рекламодавці склали значну частину рекламодавців мобільних додатків на рівні 24%.

#### 4.2 Розрахування витрат на розробку проєкту

У таблиці 4.1 зображено витрати на розробку.

Таблиця 4.1 - Кошторис витрат на проєктування

Найменування статей витрат	Сума, грн	Обґрунтування
1. Зарплата проєктувальників.	7.600	
2. Відрахування на соціальні потреби.	800	ЄСВ - 11%
3. Контрагентські роботи і послуги.	3000	39%

4. Витрати на відрядження.	0	
5. Інші прямі витрати.	1200	7.600 – 12%
6. Усього прямих витрат.	3800	800 + 3000
7. Накладні витрати.	780	3800 – 21%
8. Планові накопичення.	962	(3800 + 780) – 21%
9. Усього, кошторисна вартість проєкту.	9362	7600 + 800 + 962
10. Податок на додану вартість.	3089	9362 – 33%
11. Загалом, договірна ціна розробки ЗП.	12451	9362 + 3089

Заробітна плата працівникам розраховується на одного учасника, інженера.

Таблиця 4.2 розрахунок заробітної плати.

N	Посада	Оклад,	Відрахування	Кількість		Сума
п/п	виконавця	грн/міс	грн/міс	чол.	місяців	з/п, грн.
1				1	2	
Усього зарплати: 11000						

#### 4.3 Обґрунтування необхідності розробки

Ігровий дизайн - одна з небагатьох справжніх робочих місць, яка наповнена хвилюванням та веселощами, але це виходить за рамки цього. Індустрія постійно розвивається, представляючи нові сценарії для дизайнерів для перетасовки, і разом з цим приходить незліченна кількість нагород.

##### 1. Випадкове та все ж захоплене робоче середовище

					ДП. КН 21.433.02.000 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Культура ігрових дизайнерів складається з людей, які живуть своєю найвищою пристрастю. Через це вони захоплені, віддані справі та творчі. Вони здатні відчувати задоволення від створення чогось, чим не тільки вони особисто захоплені, але що інші також можуть насолоджуватися. Саме це допомагає зробити робоче середовище ігрового дизайну відмінним від будь-якої іншої галузі.

## 2. Ступінь не є абсолютно необхідною, але це плюс

Хоча більшість дизайнерів відеоігор отримують вищу освіту або сертифікацію, це не є абсолютно необхідним. Багато коледжів не пропонують спеціальний диплом для ігрового дизайну, тому відвідайте школи ігрового дизайну, щоб переконатися, що ви можете влаштуватися на роботу в цій галузі, взявши участь у чудовій програмі. Багато розробників наймуть дизайнера, якщо вони знають, що вони здатні створити гру; однак наявність диплому забезпечить на навчання, яке буде цінним при пошуку роботи.

## 3. Використовуйте аналітичне та творче мислення

Ті, хто розробляє ігри, несуть відповідальність за концептуалізацію багатьох частин гри, від правил до тем та структури. Є кілька спеціалізованих областей роботи, таких як дизайн рівнів, написання сценаріїв, дизайн ігор та механіка рівнів, але всі сфери використовують як мистецькі навички, так і технологічні знання.

Дуже важливо бути організованими та аналітичними при розробці відеоігор. Ці якості визначатимуть, як ви вирішуєте проблеми, наприклад, як персонаж буде реагувати на перешкоду в грі, і як гравець буде набирати очки за виконані місії.

## 4. Робіть більше, ніж просто розважайте

Відеоігри - це не просто розвага для нудьгуючих підлітків. Вони створені для допомоги в ряді реальних життєвих ситуацій, таких як депресія, стрес та проблеми з вагою. Відеоігри використовуються у навчанні солдатів, навчанні дітей, та підготовка космічної місії. Те, що колись використовувалось виключно як розвага для дітей, зараз є важливим інструментом, який допомагає покращити світ, а також допомагає деяким людям долати різні виклики.

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		



## ВИСНОВКИ

В ході дипломного проєктування, в першому розділі було проаналізовано початок розвитку індустрії мобільних ігор. Були висвітлені перші мобільні ігри в жанрі «Пригода». Була сформульована актуальність проєкту та поставлені задачі. В другому розділі були більш детально описані вимоги до системи проєкту, зокрема спроектований користувацький інтерфейс та алгоритм роботи системи. Третій розділ складався з системних вимог, реалізації структури системи, реалізації програмного забезпечення та самого тестування, яке пройшло успішно.

Вибір платформи, на якій був створений проєкт, а саме мобільна гра, зупинився на Corona Simulator. Легкий у вивченні та потужний Corona - це платформа, що ідеально підходить для швидкого створення додатків та ігор для мобільних пристроїв. Це означає, що в ній можна створити свій проєкт один раз і опублікувати його на різних типах пристроїв, включаючи Apple iPhone і iPad, телефони та планшети Android, Amazon Fire, Mac Desktop, Windows Desktop, і навіть підключені телевізори, такі як Apple TV, Fire TV та Android TV.

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Пригодницька гра. *Uk.wikipedia.org*: вебсайт. URL: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi2q6Wy5\\_xAhWLgP0HHQT7CzgQFjAAegQIBRAD&url=https%3A%2F%2Fuk.wikipedia.org%2Fwiki%2F%25D0%259F%25D1%2580%25D0%25B8%25D0%25B3%25D0%25BE%25D0%25B4%25D0%25BD%25D0%25B8%25D1%2586%25D1%258C%25D0%25BA%25D0%25B0\\_%25D0%25B2%25D1%2596%25D0%25B4%25D0%25B5%25D0%25BE%25D0%25B3%25D1%2580%25D0%25B0&usg=AOvVaw1\\_cQ5rvuNTOC9N-XIQwUux](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi2q6Wy5_xAhWLgP0HHQT7CzgQFjAAegQIBRAD&url=https%3A%2F%2Fuk.wikipedia.org%2Fwiki%2F%25D0%259F%25D1%2580%25D0%25B8%25D0%25B3%25D0%25BE%25D0%25B4%25D0%25BD%25D0%25B8%25D1%2586%25D1%258C%25D0%25BA%25D0%25B0_%25D0%25B2%25D1%2596%25D0%25B4%25D0%25B5%25D0%25BE%25D0%25B3%25D1%2580%25D0%25B0&usg=AOvVaw1_cQ5rvuNTOC9N-XIQwUux) (дата звернення: 05.01.2021).
2. Кращі 20 пригодницькі ігри для Android. *uk.wondershare.com*: вебсайт. URL: <http://uk.wondershare.com/games/best-adventure-games-android.html> (дата звернення: 09.02.2021).
3. Квест (змагання). *Uk.wikipedia.org*: вебсайт. URL: [https://uk.wikipedia.org/wiki/Квест\\_\(змагання\)](https://uk.wikipedia.org/wiki/Квест_(змагання)) (дата звернення 27.02.2021).
4. Діаграми «сутність-зв'язок» – самовчитель UML. *Waykun*: веб-сайт. URL: <https://jak.waykun.com/articles/diagrami-sutnist-zv-jazok-samovchitel-uml.html> (дата звернення: 05.03.2021).
5. Android. *Uk.wikipedia.org*: вебсайт. URL: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjsjJTn0J\\_xAhWO\\_7sIHUhBBTEQFjABegQIAxAE&url=https%3A%2F%2Fuk.wikipedia.org%2Fwiki%2FAndroid&usg=AOvVaw1sp2F7aBmxdyRIvVXuXb9w](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjsjJTn0J_xAhWO_7sIHUhBBTEQFjABegQIAxAE&url=https%3A%2F%2Fuk.wikipedia.org%2Fwiki%2FAndroid&usg=AOvVaw1sp2F7aBmxdyRIvVXuXb9w) (дата звернення: 12.03.2021).
6. Створення схем IDEF0. *support.microsoft.com*: вебсайт. URL: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj7hujG0Z\\_xAhVtposKHTgrCBIQFjABegQIBBAD&url=https%3A%2F%2Fsupport.microsoft.com%2Fukua%2Foffice%2F%25D1%2581%25D1%2582%25D0%25B2%25D0%25BE%25D1%2580%25D0%25B5%25D0%25BD%25D0%25BD%25D1%258F-%25D1%2581%25D1%2585%25D0%25B5%25D0%25BC-idef0-ea7a9289-](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj7hujG0Z_xAhVtposKHTgrCBIQFjABegQIBBAD&url=https%3A%2F%2Fsupport.microsoft.com%2Fukua%2Foffice%2F%25D1%2581%25D1%2582%25D0%25B2%25D0%25BE%25D1%2580%25D0%25B5%25D0%25BD%25D0%25BD%25D1%258F-%25D1%2581%25D1%2585%25D0%25B5%25D0%25BC-idef0-ea7a9289-)

					ДП. КН 21.433.02.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

96e0-4df8-bb26-a62ea86417fc&usg=AOvVaw0JIUCtTcZMS0njSYfNEBS0 (дата звернення: 15.03.2021).

7. Corona simulator. *corona-land.org*: вебсайт. URL: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUK EwiTzKej0p\\_xAhVIhf0HHcUECJUQFjACegQIAhAE&url=https%3A%2F%2Fcorona-land.org%2F&usg=AOvVaw30FXUONor69SBNt0kSZynC](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUK EwiTzKej0p_xAhVIhf0HHcUECJUQFjACegQIAhAE&url=https%3A%2F%2Fcorona-land.org%2F&usg=AOvVaw30FXUONor69SBNt0kSZynC) (дата звернення 29.03.2021).

8. Corona: Free Cross-Platform 2D Game Engine. *coronalabs.com*: вебсайт. URL: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUK EwiTzKej0p\\_xAhVIhf0HHcUECJUQFjADegQIBhAE&url=https%3A%2F%2Fcoronalabs.com%2F&usg=AOvVaw3sCnKJS2SX2q-46MvyOX0N](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUK EwiTzKej0p_xAhVIhf0HHcUECJUQFjADegQIBhAE&url=https%3A%2F%2Fcoronalabs.com%2F&usg=AOvVaw3sCnKJS2SX2q-46MvyOX0N) (дата звернення 04.04.2021).

9. Corona SDK. *corona-land.org*: вебсайт. URL: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUK Ewjqt6t05\\_xAhVRAxAIHcJOAGMQFjAAegQIBxAD&url=https%3A%2F%2Fcoronalabs.com%2F&usg=AOvVaw3sCnKJS2SX2q-46MvyOX0N](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUK Ewjqt6t05_xAhVRAxAIHcJOAGMQFjAAegQIBxAD&url=https%3A%2F%2Fcoronalabs.com%2F&usg=AOvVaw3sCnKJS2SX2q-46MvyOX0N) (дата звернення 13.04.2021).

10. Lua. *Uk.wikipedia.org*: вебсайт. URL: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUK EwjDjcv05\\_xAhVylYsKHTBnAGAQFjABegQIAxAD&url=https%3A%2F%2Fuk.wikipedia.org%2Fwiki%2FLua&usg=AOvVaw0DcbMubD7TyLTSq5qwKo4u](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUK EwjDjcv05_xAhVylYsKHTBnAGAQFjABegQIAxAD&url=https%3A%2F%2Fuk.wikipedia.org%2Fwiki%2FLua&usg=AOvVaw0DcbMubD7TyLTSq5qwKo4u) (дата звернення 17.04.2021).

11. Lua за 60 минут | Santa Simplicita. *zserge.wordpress.com*: вебсайт. URL: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUK EwiBlf6y1J\\_xAhV08rsIHbhBCBAQFjABegQIAhAE&url=https%3A%2F%2Fzserge.wordpress.com%2F2012%2F02%2F23%2FLua-%25D0%25B7%25D0%25B0-60-%25D0%25BC%25D0%25B8%25D0%25BD%25D1%2583%25D1%2582%2F&usg=AOvVaw2AQ5AW5nghx3bXhBDCJmxv](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUK EwiBlf6y1J_xAhV08rsIHbhBCBAQFjABegQIAhAE&url=https%3A%2F%2Fzserge.wordpress.com%2F2012%2F02%2F23%2FLua-%25D0%25B7%25D0%25B0-60-%25D0%25BC%25D0%25B8%25D0%25BD%25D1%2583%25D1%2582%2F&usg=AOvVaw2AQ5AW5nghx3bXhBDCJmxv) (дата звернення 07.05.2021).

12. 2d векторы, картинки, клипарт 2d. *ru.depositphotos.com*: вебсайт. URL: <https://ru.depositphotos.com/vector-images/2d.html> (дата звернення 23.05.2021).

					ДП. КН 21.433.02.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

## ДОДАТКИ

### Додаток А

#### Програмний код веб-застосунку

```
local vk = require("plugin_vk_direct")
--local appodeal = require( "plugin.appodeal" )

local gameStatus = 0

local yLand = display.actualContentHeight - display.actualContentHeight*0.2
local hLand = display.actualContentHeight * 0.1
local xLand = display.contentCenterX

local yBird = display.contentCenterY-50
local xBird = display.contentCenterX-50

local wPipe = display.contentCenterX+10
local yReady = display.contentCenterY-140

local uBird = -200
local vBird = 0
local wBird = -320
local g = 800
local dt = 0.025

local score = 0
local bestScore = 0
local scoreStep = 5

local bird
local land
local title
local getReady
local gameOver
local emitter

local board
local scoreTitle
local bestTitle
local silver
local gold

local pipes = {}

local function loadSounds()
```

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

```

dieSound = audio.loadSound( "Sounds/sfx_die.caf" )
hitSound = audio.loadSound( "Sounds/sfx_hit.caf" )
pointSound = audio.loadSound( "Sounds/sfx_point.aif" )
swooshingSound = audio.loadSound( "Sounds/sfx_swooshing.caf" )
wingSound = audio.loadSound( "Sounds/sfx_wing.caf" )
boomSound = audio.loadSound( "Sounds/sfx_boom.mp3" )
end

local function call_VK_event()
    local args={}
    args.user_id='33251324'
    args.activity_id=2
    args.value=score
    vk.api('secure.addAppEvent', args)
end

local function saveScoreToVk()

    if score>3 then
        call_VK_event()
    elseif score>0 then
        vk.showLeaderboardBox(score)
    else
        vk.showShareBox("I just scored " .. score .. "! Create your own
game with Corona.", {"https://coronalabs.com/", }, "wall")
    end
end

local function calcRandomHole()
    return 100 + 20*math.random(10)
end

local function loadBestScore()
    local path = system.pathForFile( "bestscore.txt", system.Document
sDirectory )

    -- Open the file handle
    local file, errorString = io.open( path, "r" )

    if not file then
        -- Error occurred; output the cause
        print( "File error: " .. errorString )
    else
        -- Read data from file
        local contents = file:read( "*a" )
        -- Output the file contents
        bestScore = tonumber( contents )
    end
end

```

					ДП. КН 21.433.02.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        -- Close the file handle
        io.close( file )
    end

    file = nil
end

local function saveBestScore()
    -- Path for the file to write
    local path = system.pathForFile( "bestscore.txt", system.Document
sDirectory )
    local file, errorString = io.open( path, "w" )
    if not file then
        -- Error occurred; output the cause
        print( "File error: " .. errorString )
    else
        file:write( bestScore )
        io.close( file )
    end
    file = nil

    -- show appodeal ad
    -- appodeal.show()

end

local function setupBird()
    local options =
    {
        width = 70,
        height = 50,
        numFrames = 4,
        sheetContentWidth = 280, --
- width of original 1x size of entire sheet
        sheetContentHeight = 50 --
- height of original 1x size of entire sheet
    }
    local imageSheet = graphics.newImageSheet( "Assets/bird.png", opt
ions )

    local sequenceData =
    {
        name="walking",
        start=1,
        count=3,
        time=300,
    }

```

					<i>ДП. КН 21.433.02.000 ПЗ</i>	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        loopCount = 2,      -
- Optional ; default is 0 (loop indefinitely)
        loopDirection = "forward"      -
- Optional ; values include "forward" or "bounce"
    }
    bird = display.newSprite( imageSheet, sequenceData )
    bird.x = xBird
    bird.y = yBird
end

local function prompt(tempo)
    bird:play()
end

local function initGame()
    score = 0
    scoreStep = 5
    title.text = score
--    title.text = hLand

    for i=1,3 do
        pipes[i].x = 400 + display.contentCenterX * (i-1)
        pipes[i].y = calcRandomHole()
    end
    yBird = display.contentCenterY-50
    xBird = display.contentCenterX-50
    getReady.y = 0
    getReady.alpha = 1
    gameOver.y = 0
    gameOver.alpha = 0
    board.y = 0
    board.alpha = 0
    audio.play( swooshingSound )
    transition.to( bird, { time=300, x=xBird, y=yBird, rotation = 0 }
)
    transition.to( getReady, { time=600, y=yReady, transition=easing.
outBounce, onComplete=prompt    } )
end

local function wing()
    if gameStatus==0 then
        gameStatus=1
        getReady.alpha = 0
    end

    if gameStatus==1 then
        vBird = wBird
        bird:play()
    end
end

```

```

        audio.play( wingSound )
    end

    if gameStatus==3 then
        gameStatus=0
        initGame()
    end
end

local function  setupExplosion()
    local dx = 31
    local p = "Assets/habra.png"
    local emitterParams = {
        startParticleSizeVariance = dx/2,
        startColorAlpha = 0.61,
        startColorGreen = 0.3031555,
        startColorRed = 0.08373094,
        yCoordFlipped = 0,
        blendFuncSource = 770,
        blendFuncDestination = 1,
        rotatePerSecondVariance = 153.95,
        particleLifespan = 0.7237,
        tangentialAcceleration = -144.74,
        startParticleSize = dx,
        textureFileName = p,
        startColorVarianceAlpha = 1,
        maxParticles = 128,
        finishParticleSize = dx/3,
        duration = 0.75,
        finishColorRed = 0.078,
        finishColorAlpha = 0.75,
        finishColorBlue = 0.3699196,
        finishColorGreen = 0.5443883,
        maxRadiusVariance = 172.63,
        finishParticleSizeVariance = dx/2,
        gravityy = 220.0,
        speedVariance = 258.79,
        tangentialAccelVariance = -92.11,
        angleVariance = -300.0,
        angle = -900.11
    }
    emitter = display.newEmitter(emitterParams )
    emitter:stop()
end

local function explosion()
    emitter.x = bird.x
    emitter.y = bird.y
    emitter:start()
end

```

					ДП. КН 21.433.02.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		



end

```
local function crash()
    gameStatus = 3
    audio.play( hitSound )
    gameOver.y = 0
    gameOver.alpha = 1
    transition.to( gameOver, { time=600, y=yReady, transition=easing.
outBounce } )
    board.y = 0
    board.alpha = 1
```

```
saveScoreToVk()
```

```
if score>bestScore then
    bestScore = score
    saveBestScore()
end
bestTitle.text = bestScore
scoreTitle.text = score
if score<10 then
    silver.alpha = 0
    gold.alpha = 0
elseif score<50 then
    silver.alpha = 1
    gold.alpha = 0
else
    silver.alpha = 0
    gold.alpha = 1
end
transition.to( board, { time=600, y=yReady+100, transition=easing
.outBounce } )
end
```

```
local function collision(i)
    local dx = 40 -- horizontal space of hole
    local dy = 50 -- vertical space of hole
    local boom = 0
    local x = pipes[i].x
    local y = pipes[i].y

    if xBird > (x-dx) and xBird < (x+dx) then
        if yBird > (y+dy) or yBird < (y-dy) then
            boom = 1
```

					ДП. КН 21.433.02.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        end
    end
    return boom
end

local function gameLoop()
    local eps = 10
    local leftEdge = -60
    if gameStatus==1 then
        xLand = xLand + dt * uBird
        if xLand<0 then
            xLand = display.contentCenterX*2+xLand
        end
        land.x = xLand
        for i=1,3 do
            local xb = xBird-eps
            local xOld = pipes[i].x
            local x = xOld + dt * uBird
            if x<leftEdge then
                x = wPipe*3+x
                pipes[i].y = calcRandomHole()
            end
            if xOld > xb and x <= xb then
                score = score + 5
                title.text = score .. " метрів"
                if score==scoreStep then
                    scoreStep = scoreStep + 5
                    audio.play( pointSound )
                end
            end
            pipes[i].x = x
            if collision(i)==1 then
                explosion()
                audio.play( dieSound )
                gameStatus = 2
            end
        end
    end
end

if gameStatus==1 or gameStatus==2 then
    vBird = vBird + dt * g
    yBird = yBird + dt * vBird
    if yBird>yLand-eps then
        yBird = yLand-eps
        crash()
    end
    bird.x = xBird
    bird.y = yBird
    if gameStatus==1 then
        bird.rotation = -30*math.atan(vBird/uBird)
    end
end

```

```

        else
            bird.rotation = vBird/8
        end
    end
end
end

local function setupLand()
    land = display.newImageRect( "Assets/land.png", display.actualCon
tentWidth*2, hLand*2 )
    land.x = xLand
    land.y = yLand+hLand
end

local function setupImages()
    local ground = display.newImageRect( "Assets/ground.png", display
.actualContentWidth, display.actualContentHeight )
    ground.x = display.contentCenterX
    ground.y = display.contentCenterY
    ground:addEventListener("tap", wing)

    for i=1,3 do
        pipes[i] = display.newImageRect( "Assets/pipe.png", 80, 1000 )
        pipes[i].x = 440 + wPipe * (i-1)
        pipes[i].y = calcRandomHole()
    end

    getReady = display.newImageRect( "Assets/getready.png", 200, 60 )
    getReady.x = display.contentCenterX
    getReady.y = yReady
    getReady.alpha = 0

    gameOver = display.newImageRect( "Assets/gameover.png", 200, 60 )
    gameOver.x = display.contentCenterX
    gameOver.y = 0
    gameOver.alpha = 0

    board = display.newGroup()
    local img = display.newImageRect(board, "Assets/board.png", 240,
140 )

    scoreTitle = display.newText(board, score, 80, -
18, "Assets/troika.otf", 21)
    scoreTitle:setFillColor( 0.75, 0, 0 )
    bestTitle = display.newText(board, bestScore, 80, 24, "Assets/tro
ika.otf", 21)
    bestTitle:setFillColor( 0.75, 0, 0 )

    silver = display.newImageRect(board, "Assets/silver.png", 44, 44
)
    silver.x = -64

```

					ДП. КН 21.433.02.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

```

silver.y = 4

gold = display.newImageRect(board, "Assets/gold.png", 44, 44 )
gold.x = -64
gold.y = 4

board.x = display.contentCenterX
board.y = 0
board.alpha = 0

local txt = {
    x=display.contentCenterX, y=60,
    text="",
    font="Assets/troika.otf",
    fontSize=35 }

title = display.newText(txt)
title:setFillColor( 1, 1, 1 )
end

local function vkListener( event )
--         if event.status == "success" then
--             loadingText.text = event.method
--         end
end

-- Start application point
loadSounds()
setupImages()
setupBird()
setupExplosion()
setupLand()
initGame()
loadBestScore()
gameLoopTimer = timer.performWithDelay( 25, gameLoop, 0 )

-- debug text line
local loadingText = display.newText( "Debug info", display.contentC
enterX, display.contentCenterY, nil, 20)

-- vk listener
local function vkListener( event )
    loadingText.text = "version 1229\nevent = " .. event.method
    if event.method == 'init' then
        if event.status == 'success' then
            loadingText.text = ""
        end
    end
--     loadingText:removeSelf( )

```

```

        else
            loadingText.text = "Error while loading\n(" .. tostring(event
            .data and event.data.message) .. ")"
        end
    end
end

-- appodeal listener
local function adListener( event )
    if ( event.phase == "init" ) then -- Successful initialization
        -
        - maybe set a flag that you can see in all scenes to know that init
        ialization is complete

        elseif ( event.phase == "failed" ) then -
        - The ad failed to load
            print( event.type )
            print( event.isError )
            print( event.response )
        end
    end

display.setStatusBar( display.HiddenStatusBar )

if system.getInfo('platform') ~= 'html5' then
    timer.performWithDelay( 100, function( )
        loadingText.text = "~html5"
        vkListener{ method = 'init', status='success' }
    end )
else
    loadingText.text = "html5"
    vk.init(vkListener)
end

--appodeal.init( adListener,
-
- { appKey="d0b151949cc7fdaecc358106bb00d606fdc7d51b70436d36",
--     locationTracking = false,
--     supportedAdTypes = {"interstitial"},
--     childDirectedTreatment = true,
--     bannerAnimation = true,
--     testMode = true } )

```