

## ЗМІСТ

ЗМІСТ.....	3
ВСТУП.....	4
1 ТЕОРЕТИЧНІ ВІДОМОСТІ .....	5
1.1 Аналіз предметної області та аналогів .....	5
1.2 Аналіз технологій та засобів розробки.....	8
1.3 Висновки.....	9
2 ТЕХНІЧНЕ ЗАВДАННЯ.....	11
3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ .....	15
3.1 Проектування архітектури .....	15
3.2 Розробка макетів .....	15
3.3 Реалізація .....	17
3.4 Тестування .....	20
3.5 Інструкція користувача .....	21
3.6 Висновок .....	23
ВИСНОВОК .....	25
ПЕРЕЛІК ПОСИЛАНЬ.....	26
Додаток А.....	27
Додаток Б.....	28

					ІС КР 6.050101 АІ151 ПЗ			
<b>Змі</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>	Розробка мобільної гри “Tower Crush”	<b>Літ.</b>	<b>Лист</b>	<b>Листів</b>
Розроб.		Унтілов А.А.						
Перев.		Маковецький О.С.					3	26
Реценз.						ОНПУ, каф. ІС, гр. АІ151		
Н. Контр.								
Утверд.								

## ВСТУП

В наш час з підвищенням обчислювальної спроможності кишенькових пристроїв ( мобільних телефонів), люди мають практично безмежні можливості щодо роботи з ними. Телефони дозволяють отримувати будь-яку інформацію де б не була людина, редагувати файли, орієнтуватися на місцевості, прослуховувати музика і т.п. Тепер можливо навіть грати в сучасні ігри, які раніше потребували великої на той час кількості ресурсів пристрою. В наш час люди постійно знаходяться в русі, тому їм потрібні ігри, в які б вони могли грати де завгодно, коли завгодно: користуючись громадським транспортом, сидячі в черзі і т.д. Саме тому було вирішено розробити просту, але цікаву гру для мобільних пристроїв, які зараз користуються великою популярністю, та в яку люди захотіли б грати у час очікування та в дорозі.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати завдання і сучасні підходи до створення мобільних ігор на прикладі декількох аналогів;
- виробити вимоги до розробленої гри;
- провести аналіз і обґрунтувати вибір технологій для створення гри;
- провести проектування загальної структури гри, структури її окремих модулів;
- здійснити програмну реалізацію гри на підставі отриманої інформації на етапі проектування;
- провести тестування розробленої гри.

# 1 ТЕОРЕТИЧНІ ВІДОМОСТІ

## 1.1 Аналіз предметної області та аналогів

Гра “Tower Crush” являє собою просту аркадну гру у якій гравцеві необхідно ламати нижні блоки башти, уникаючи перешкоди, які випадковим шляхом з’являються зліва і справа од неї. Щоб контролювати головного героя, гравцю необхідно обрати з якої сторони він зламає наступний блок башти і натиснути на відповідну частину екрану. Головна мета гравця - набрати якомога більше очок, які здобуваються за кожний зламаваний блок вежі. Після гри гравцеві надається інформація про набрану їм кількість очок.

Під час аналізу предметної області були розглянуті наступні аналоги.

Перший аналог – гра “Timberman”.



Рисунок 1.1– Головне меню гри “Timberman”

Гра виконана у піксельному стилі старих ігрових автоматів, головне меню містить кнопку старту гри, таблиці рекордів і вибору зовнішнього вигляду(костюму) гравця. Основна ціль гри -зрубати дерево, доки шкала верху екрану не дійде до нуля, крім того під час зрубання на гравця

надвигаються перешкоди, які йому необхідно минати.

Під час аналізу гри було виявлено, що вона:

- має приємний дизайн;
- не має особливих режимів гри;
- не має можливості змінити ігрову локацію.

Другий аналог – гра “Tower Boxing” .



Рисунок 1.2 – Гра “Tower Boxing”

Порівняно з першим аналогом гра має більш складне головне меню, з можливістю підключення аккаунтів facebook і tweeter, має вікно achievements. Крім того у ігровий процес був доданий процес збору ігрової валюти, за яку можна купувати нові костюми гравця.

Під час аналізу гри було виявлено, що вона:

- головне меню має більше функцій;
- у ігровому процесі з'явився аспект збору ігрової валюти;

					ІС КР 6.050101 АІ151 ПЗ	Лист
						6
Змін	Лист	№ докум.	Підпис	Дата		

- все ще доступна тільки 1 ігрова локація.

Третій аналог – гра “Totem smash”.

У порівнянні з попередніми аналогами гра має більш сучасну графічну частину, має різні локації гри, які змінюються випадково. Проте у грі не має можливості купівлі нових костюмів та відсутня ігрова валюта. Головне меню гри дуже схоже з 1 аналогом і має такі ж самі пункти.

Під час аналізу гри було виявлено, що вона:

- має більш сучасну деталізовану графіку;
- має різні анімовані фони рівнів;
- не має валюти або магазину у будь-якому вигляді.



Рисунок 1.3 – Головне меню гри “Totem smash”.

Після аналізу усіх аналогів було прийнято рішення розробити просту аркадну гру, яка не буде потребувати від гравця якихось особливих навичок. Розробити гарний, зрозумілий інтерфейс і дати гравцю невелику інструкцію

щодо управління. Ідея гри дуже проста: дракон ламає башту лицарів, які займаються охотою на драконів, тим самим намагається зберегти себе і свій рід. Завдяки такому незвичному головному герою гра вирізняється з існуючих аналогів і має свою неповторну атмосферу.

## 1.2 Аналіз технологій та засобів розробки

У даній роботі ставилося завдання вибрати середовище розробки і фреймворк, які підтримують вже вивчену мову Java, а також мають можливості створення кросплатформних додатків. Було прийняте рішення розробити гру для мобільних девайсів на ОС Android, оскільки виконавець проекту має саме такий телефон і на ньому можна буде вдало проводити тестування гри під час її розробки.

Оскільки треба розробити просту 2D гру без 3вимірної графіки та фізики, для її розробки був обраний популярний фреймворк для створення мобільних ігор і додатків - LibGDX.

LibGDX – це кросплатформний відкритий движок на Java, що дозволяє створювати 2D і 3D ігри під різні платформи, серед яких: Android, HTML5, Windows, Linux, Mac OS X.

LibGDX дозволяє розроблювачеві писати, тестувати і налагоджувати код на власному комп'ютері і потім переносити його на інші ОС. Для цього фреймворк використовує окремі модулі для складання додатка під кожную платформу, а так само незалежний модуль, який містить основний код додатка.

- Платформа: Android, HTML5, Windows, Linux, Mac OS X;
- Мови програмування: Java;
- Переваги: безкоштовний; підтримка Windows, Linux, Mac OS X,

Android ОС, браузери з підтримкою WebGL і iOS. 32 і 64-розрядні версії ОС; гнучкість і збільшення можливостей досягається за рахунок розширень, можна підключити фізичний движок Box2D для роботи з об'єктами і фізикою реального миру, додати підтримку TrueType шрифтів або працювати з 3D об'єктами; вбудований набір віджетів для малювання графічного інтерфейсу – як ігор так і програм; наявність можливості легко реалізувати логікові поведінки часток; висока продуктивність.

Для роботи з LibGDX було обрано Android Studio – інтегроване середовище розробки (IDE) для роботи із платформою Android. Android Studio, заснована на програмному забезпеченні IntelliJ IDEA від компанії JetBrains, є офіційним засобом розробки Android додатків. Дане середовище розробки доступне для Windows, OS X і Linux. використовує мови програмування Java і C++. Крім того у виконавця проекту вже є досвід праці з даним середовищем розробки.

### 1.3 Висновки

У цьому розділі було проведено аналіз предметної області та аналогів, аналіз технологій та засобів розробки. Під час аналізу предметної області були виявлені та сформовані вимоги до розроблюваного мобільної гри, а також були розглянуті приклади ігор-аналогів, в яких було виділено певні особливості, переваги та недоліки кожного з аналогів. Після аналізу були виявлені певні об'єкти та технології, які плануються для використання у грі – простий і зрозумілий інтерфейс, невелика інструкція щодо керування, таблиця результатів гравця та його найкращого рахунку.

Під час аналізу технологій та засобів розробки були розглянуті сучасні технології для мобільних ігор. Для даного проекту були вибрані JAVA в якості

					ІС КР 6.050101 АІ151 ПЗ	Лист
Змін	Лист	№ докум.	Підпис	Дата		9

язика програмування, Android Studio в якості середовища розробки та libGDX у якості основного фреймворку.

					ІС КР 6.050101 АІ151 ПЗ	Лист
						10
Змін	Лист	№ докум.	Підпис	Дата		



## 2 ТЕХНІЧНЕ ЗАВДАННЯ

Додаток №1:

Технічне завдання на розробку мобільної гри “Tower Crush”.

### 1. Введення

#### 1.1. Найменування програми

Найменування гри: “Tower Crush”.

#### 1.2. Призначення і область застосування

Гра призначена для роботи на мобільних девайсах з ОС Android.

### 2. Вимоги до програми

#### 2.1. Нефункціональні вимоги

Гра повинна бути графічно привабливою, мати простий інтерфейс та цікавий геймплей.

#### 2.2. Функціональні вимоги

Гра повинна забезпечувати можливість виконання перерахованих нижче функцій:

##### 2.2.1. Можливість почати нову гру

##### 2.2.2. Можливість керування головним персонажем

##### 2.2.3. Можливість перегляду фінального рахунку гравця після гри

##### 2.2.4. Перегляд таблиці рекордів

#### 2.3. Вимоги до надійності

##### 2.3.1 Вимоги до забезпечення надійного функціонування програми

Надійне (стійке) функціонування гри повинно бути забезпечено виконанням необхідних технічних вимог.

					ІС КР 6.050101 АІ151 ПЗ	Лист
						11
Змін	Лист	№ докум.	Підпис	Дата		

### 3.2. Вимоги до складу і параметрів технічних засобів

До складу технічних засобів користувача повинен входити мобільний телефон з наступними характеристиками:

Мінімальні необхідні вимоги:

- Дісплей 5”
- Оперативна пам'ять 1 ГБ
- Внутрішня пам'ять 4 ГБ
- Вільна пам'ять 100 мб
- Процесор 4 ядра по 1.2 ГГц

### 3.3. Вимоги до інформаційної та програмної сумісності

#### 3.3.1. Вимоги до інформаційних структур і методів розв'язання

Особливих вимог до інформаційних структур програма не пред'являє.

#### 3.3.2. Вимоги до вихідного коду і мов програмування

Для розробки вихідних кодів буде використовуватися IDE Android Studio

В якості мови розробки буде використана Java.

#### 3.3.3. Вимоги до програмних засобів, які використовуються програмою

Потребує версію ОС Android 4.3 і вище.

#### 3.4.4. Вимоги до захисту інформації та програм

Вимоги до захисту інформації та програм не пред'являються.

### 4. Вимоги до програмної документації

#### 4.1. Попередній склад програмної документації

Склад програмної документації повинен включати в себе:

- 4.1.1. технічне завдання;
- 4.1.2. програму і методики випробувань.

### 5. Техніко-економічні показники

#### 5.1. Економічні переваги розробки

					ІС КР 6.050101 АІ151 ПЗ	Лист
Змін	Лист	№ докум.	Підпис	Дата		12

Орієнтовна економічна ефективність не розраховуються.

## 6. Стадії та етапи розробки

### 6.1. стадії розробки

Розробка повинна бути проведена в три стадії:

1. розробка технічного завдання;
2. розробка програмного забезпечення;
3. використання.

### 6.2. Етапи розробки

На стадії розробки технічного завдання повинен бути виконаний етап розробки, узгодження і затвердження цього технічного завдання.

На стадії розробки програмного забезпечення повинні бути виконані перераховані нижче етапи робіт:

1. проектування гри;
2. реалізація гри;
3. розробка програмної документації;
4. випробування гри.

На стадії впровадження, готовий програмний продукт повинен бути встановлений на мобільному девайсі користувача.

### 6.3. Зміст робіт по етапах

На етапі розробки технічного завдання повинні бути виконані перераховані нижче роботи:

1. постановка задачі;
2. визначення та уточнення вимог до технічних засобів;
3. визначення вимог до програми;
4. визначення стадій, етапів і термінів розробки програми і документації на неї;

5. узгодження і затвердження технічного завдання.

На етапі розробки програми повинна бути виконана робота з проектування, програмування (кодування) і налагодженні програми.

На етапі розробки програмної документації повинна бути виконана розробка програмних документів відповідно до вимог до складу документації.

На етапі випробувань програми повинні бути виконані перераховані нижче види робіт:

1. розробка, погодження та затвердження і методики випробувань;
2. проведення приймально-здавальних випробувань;
3. коригування програми і програмної документації за результатами випробувань.

На етапі впровадження програма повинна бути передана Замовнику, після чого він може почати її використання.

#### 7. Порядок контролю і приймання

##### 7.1. Види випробувань

Приймально-здавальні випробування повинні проводитися на об'єкті Замовника в обумовлені терміни.

Приймально-здавальні випробування програми повинні проводитися згідно з розробленою Виконавцем і узгодженої Замовником Програми та методик випробувань.

##### 7.2. Загальні вимоги до приймання роботи

На підставі Протоколу проведення випробувань Виконавець спільно з Замовником підписує Акт приймання-здачі програми в експлуатацію.

## 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

### 3.1 Проектування архітектури

Архітектура гри базується на основі первісного проекту фреймворка libGdx, нас цікавлять наступні об'єкти:

1. 3 Основних класа:

- MyGdxGame, який є нашим основним класом і відповідає за усе, що є у грі і навіть більше.
- AndroidLauncher - відповідає за відтворення гри на платформі Android
- DesktopLauncher - відповідає за відображення нашої гри на робочому столі комп'ютера.

2. Папка assets, яка міститься в каталозі Android`а. В цій папці знаходяться всі матеріали, які необхідні для нашої гри: зображення, font`s, звуки і таке інше

3. Файл AndroidManifest.xml - містить усі параметри запуску нашої гри на мобільних девайсах.

Для проектування гри також були розроблені 2 діаграми:

- Use-case діаграма (Додаток А), яка показує можливі взаємодії гравця з грою.
- Діаграма класів (Додаток Б), яка ілюструє класи гри.

### 3.2 Розробка макетів

Для більш продуктивної роботи по створенню якісного програмного продукту необхідно спочатку уявити, як цей продукт має виглядати по завершенню роботи. Для цього необхідно побудувати модель / макет кінцевої програми.

Для проектування моделі було використано такі програми, як Pencil та Adobe Photoshop. Отримані начерки являються лише приблизним уявленням

					ІС КР 6.050101 АІ151 ПЗ	Лист
Змін	Лист	№ докум.	Підпис	Дата		15

кінцевого продукту та не зобов'язують повного дотримання в усіх деталях. Створені макети проілюстровано нижче: (рис. 3.1) та (рис. 3.2).

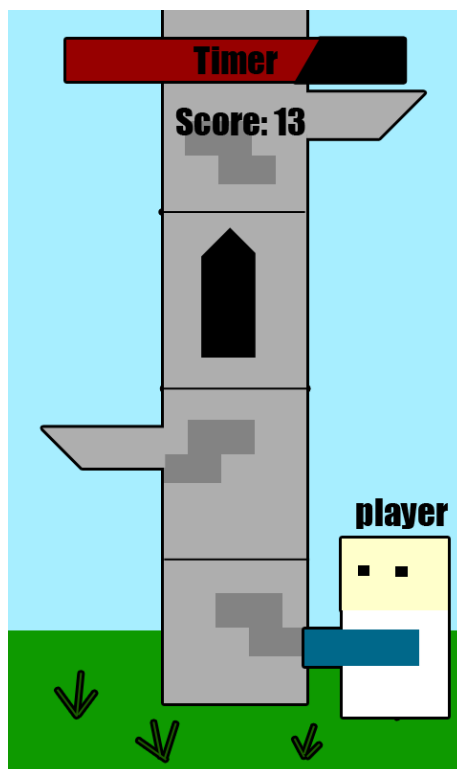


Рисунок 3.1 - Вікно ігрового процесу



Рисунок 3.2 - Вікно кінця гри

Отже були побудовані макети, які допоможуть більш точно конкретизувати кінцевий продукт. Тепер стало зрозуміло, як повинна виглядати гра, для того, щоб реалізувати даний макет, нам необхідно:

1. Створити головну сцену та накласти на неї необхідний фон
2. Створити головного персонажа гри, у нашому випадку - дракона, котрий зможе взаємодіяти з іншим персонажем гри - баштою.
3. Створити башту, яку головний герой повинен зламати, уникаючи перешкоди.
4. Додати стрічку таймера, яка з плином часу буде зменшуватися; поповнюється при натисканні гравця на екран.
5. Додати лічильник рахунку гравця, який буде збільшуватися при кожному натисканні і скидуватися до 0 наприкінці гри.
6. Додати кінцевий рахунок гравця, який буде з'являтися після кінця гри.

### 3.3 Реалізація

На цьому етапі ми будуємо основні компоненти гри - класи нашої програми. Щоб створити будь-який проект за допомогою LibGdx, спочатку необхідно створити порожній проект, це можна зробити за допомогою спеціальної програми gdx-setup.jar. На наступному зображенні (рис 3.3) ми бачимо початкові налаштування нашого проекту.

Після створення нового проекту, його необхідно імпортувати в обрану середу розробки, у нашому випадку Android Studio. Для цього необхідно зробити наступні кроки File -> Open -> Import і обираємо наш створений проект.

Розглянемо ближче основні методи нашого головного класу MyGdxGame:

- create()- створення усіх об'єктів гри та їх ініціалізація;
- render()- цей метод відповідає за малювання кожного кадру, він - викликається 60 разів в секунду.
- dispose()- знищення усіх об'єктів, викликається при виході з гри.

Додаткові створені класи та їх функції:

- 1) `GameScene` - якщо `MyGdxGame` це наша сцена, на якій відбувається вистава, то `GameScene` - наша п'єса, актори та декорації. У цьому класі розміщена уся логіка гри, у ньому створюються та займають своє місце зображення, саме з цим класом взаємодіє гравець.
- 2) `Player` - клас, який містить всі відомості про гравця: його позицію, його статус та ін.
- 3) `Wall` - клас, що відповідає за об'єкт блоку башти. Містить у собі інформацію про нього: позицію у просторі, тип.
- 4) `Walls` - наша башта це набір блоків, з яких вона складається. Цей клас є списком об'єктів класу `Wall`. Видаляє об'єкт коли гравець рушить рівень башти та додає новий зверху. Також містить у собі логіку спавну нових рівнів башти.

Логіку роботи гри можна описати наступним чином. Коли гра починається випадковим чином з'являється башта, що складається з 6 блоків і розміщена у списку класу `Walls` чергою. Коли гравець ламає нижній блок вежі, то зі списку видаляється перший елемент і відповідно до логіки спавну з'являється новий з кінця. Гра закінчується у 2 випадках:

- 1) Гравець зіштовхується з перешкодою башти, те, коли це відбувається, перевіряється логікою гри у методі `update()`.
- 2) Таймер, розміщений зверху екрана добігає кінця.

Будь-які дії гравця підвищують значення полоски таймеру. Гравець у свою чергу може обрати лише 1 річ: з якої сторони йому зламати наступний блок башти так, щоб на нього не впала перешкода з наступного рівня.





Рисунок 3.3 - Вікно створення проекту та початкові налаштування

### 3.4 Тестування

При першому тестуванні ми перевіряємо усі функції, які повинна виконувати програма.

№	Функція	Сценарій	Очікуємий результат	Результат	Підсумок
1	Запуск гри	Запустіть гру	Гра запустилась	Гра запустилась	+
2	Керування персонажем	1.Натисніть ліву частину екрану	Дракон з'явився з лівої сторони	Дракон з'явився з лівої сторони	+
		2.Натисніть праву частину екрану	Дракон з'явився з правої сторони	Дракон з'явився з правої сторони	+
3	Зіткнення з перешкодою	Зіштовхніть дракона з перешкодою	Game Over	Game Over	+
4	Закінчення таймеру	Нічого не робіть, доки таймер не скінчиться	Game Over	Game Over	+
5	Накопичення очок	Натисніть на екран 1 раз	Показник Score додасть до себе 1 пункт	Показник Score додав до себе 1 пункт	+
6	Можливість почати гру з початку	При завершенні гри натисніть на екран 1 раз	Гра почнеться знову	Гра почалась знову	+

На другому тестуванні гру будуть тестувати цільова аудиторія. Програмний продукт не вимагає конкретної вікової або проф. групи. Тому мою гру протестували три людини.

1. Дев'ятнадцятирічний студент з мого потоку. Він зауважив, що гра йому видається непоганою, йому сподобалася ідея та інтерфейс. Ігровий процес йому здався занадто легким і він попросив збільшити рівень складності.
2. Вісімнадцятирічна студентка мала проблеми з розумінням управління у грі і їй знадобився деякий час, аби до нього звикнути. Крім цього вона зауважила, що в грі не вистачає косметичних послуг.
3. Останній тестувальник вирішив zostатися інкогніто і не повідомив свій вік. Погравши у гру 15 хвилин і не змігши побити мій рекорд він зауважив, що гра занадто складна і щоб навчитися в неї грати, потрібен деякий час.

Після проведеного тестування і опросу, було виявлено, що тестувальникам загалом сподобалася концепція гри та її реалізація. Були деякі зауваження щодо складності гри, проте складність є дуже суб'єктивним показником і щоб його збалансувати, треба проводити більш детальні дослідження. Ще одним мінусом гри виявилася відсутність косметичних послуг зміни текстури гравця або рівня. На жаль зараз не має можливості додати до гри магазин через брак часу до дедлайну проекту. Проте у майбутніх доповненнях до гри буде доданий такий аспект. Гра постійно розвивається і розробники постійно дослуховуються до думки своїх гравців і змінюють проект у кращу сторону.

### 3.5 Інструкція користувача

Для запуску гри на мобільному девайсі, або де інде, спершу необхідно встановити гру за допомогою файлу формату .apk. Після цього необхідно запустити гру, натиснувши на її ярлик, утворений на робочому столі.

					ІС КР 6.050101 АІ151 ПЗ	Лист
Змін	Лист	№ докум.	Підпис	Дата		21



Рисунок 3.4 -Початковий екран гри

Користувач опиниться на початковому екрані (рис 3.6). На ньому ми бачимо логотип гри, найбільший рекорд та загальну кількість набраних очок. Щоб розпочати гру, необхідно натиснути по будь-якому місцю екрана.

Головна мета гравця - набрати максимальну кількість очок, для цього необхідно ламати башту, уникаючи перешкоди, що з'являються випадковим чином зправа та зліва від башти. Також гравцю необхідно пильнувати полосу зверху екрана. Якщо гравець буде грати надто повільно - то шкала дійде до 0 і гра буде завершена. Якщо дуже швидко: збільшується шанс натрапити на перешкоду.

Змін	Лист	№ докум.	Підпис	Дата

Якщо гра буде закінчена, то посеред екрану буде написано Game Over і написаний кінцевий рахунок гравця та мотиваційна стрічка. (рис. 3.7)



Рисунок 3.5 - Екран Game Over

### 3.6 Висновок

Завдяки змодельованим макетам гри, була створена мобільна гра “Tower Crush”. Як і планувалося, були створені наступні об’єкти: головна сцена, гравець, башта, таймер, лічильник рахунку гравця;була розроблена логіка їх взаємодії усередині проекту. Програмний продукт можна вважати закінченим, проте завдяки використаним ООП парадигмам під час його

створення, його можна постійно розширювати(додавати нові елементи інтерфейсу, нові моделі гравців чи нові види перешкод) та змінювати, при цьому не порушуючи цілостності гри.

					ІС КР 6.050101 АІ151 ПЗ	Лист
Змін	Лист	№ докум.	Підпис	Дата		24

## ВИСНОВОК

У ході виконання курсової роботи, було виявлено та проаналізовано вимоги до системи, була створена UML-діаграма варіантів використання та діаграма класів. Була здійснена програмна реалізація на мові Java у середі розробки Android Studio за допомогою фреймворка LibGDX. Була розроблена гра для мобільних девайсів на ОС Android.

Був проведений аналіз аналогічних ігор. У них були виявлені позитивні і негативні аспекти, які вплинули на реалізацію програмного продукту. Також було проведено тестування програмного продукту. Усі заплановані функції були реалізовані і працюють належним чином.

Отриманий програмний продукт не є взірцем якості, проте має потенціал для подальшого покращення і розвитку. У майбутньому планується додати сюжетну лінію та бескінечний режим(який є зараз основним), та додати магазин з новими моделями головного гравця, які будуть отримуватися за досягнення у грі.

					ІС КР 6.050101 АІ151 ПЗ	Лист
Змін	Лист	№ докум.	Підпис	Дата		25

## ПЕРЕЛІК ПОСИЛАНЬ

1. Rouse, Richard. Game Design: Theory & Practice. – Los RiOS Boulevard, Plano, Texas, USA : Wordware Publishing, 2004. – 698 с.
2. Moore, Michael E. Basics of Game Design. – New York, USA : CRC Press, 2011. – 376 с.
3. Schwab, Brian. AI Game Engine Programming. – Canada : Course Technology, 2009. – 710 с.
4. Kent L., Steven. The Ultimate History Of Video Games : [англ.]. – 1. – New York : Three Rivers Press, 2001. – 608 с.
5. Бондаренко В. А., Попов М. В. К вопросу о рассмотрении компьютерных игр в качестве основы реализации виртуальных благ // Научно-методический электронный журнал «Концепт». – 2017. – Т. 31. – С. 266–270. [Электронный ресурс] - <http://e-koncept.ru/2017/970068.htm>.
6. Язев Юрій: «Обзор самых популярных движков для разработки игр». [Электронный ресурс] - <https://xakep.ru/2014/09/05/game-development-enginesreview/>
7. Вікіпедія: «Timber Man» [Електронний ресурс] - [https://ru.wikipedia.org/wiki/Timber\\_Man](https://ru.wikipedia.org/wiki/Timber_Man).
8. Аркадна гра онлайн «Tower Boxing» [Електронний ресурс] - [http://www.onlinegamesbazar.ru/game/tower\\_boxing](http://www.onlinegamesbazar.ru/game/tower_boxing).
9. Аркадна гра онлайн «Totem smash!» [Електронний ресурс] - <https://games.softgames.com/games/totemsmash/gamesites/5/locale/ru>.