МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки
Кафедра Автоматизованих систем моніторингу навколишнього середовища

КОМПЛЕКСНА МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Розробка та оптимізація сюжетної побудови мобільного додатка Dino`s Adventure на платформі Android

Склад:

Частина 1. Розробка та оптимізація сюжетної побудови мобільного додатка Dino`s Adventure

Виконав: студент групи МІС-18 Полєвой Є.В.
(п.і.б.)

Керівник: Перелигін Б.В..
(п.і.б.)

Частина 2. Розробка та оптимізація програмного комплексу мобільного додатка Dino`s Adventure

Виконав: студент групи МІС-18 Клименко М.О
(п.і.б.)

Керівник: Перелигін Б.В.
(п.і.б.)

Староста роботи: Полєвой Є.В.
(п.і.б.)

Керівник роботи: к.т.н., доцент, Перелигін Б.В.
(п.і.б.)

Рецензент: к.ф-м.н. проф., Козловська В.П.
(п.і.б.)

ОДЕСА 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки

Кафедра Автоматизованих систем моніторингу навколишнього середовища

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Розробка та оптимізація сюжетної побудови мобільного додатка Dino`s Adventure

Виконав студент 2 курсу групи МІС-18
спеціальності 122 Комп'ютерні науки

Полєвой Євген Володимирович

Керівник к.т.н., доцент.
Перелигін Б.В.

Консультант

Рецензент к.ф.-м.н., доцент
Козловська Валентина Петрівна

Одеса 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки
Кафедра Автоматизованих систем моніторингу навколишнього середовища
Рівень вищої освіти магістр
Спеціальність 122 Комп'ютерні науки
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри АСМНС
Перелигін Б.В.
" 28 " жовтня 2019 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ПОЛЄВОЙ ЄВГЕН ВОЛОДИМИРОВИЧ
(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка та оптимізація сюжетної побудови мобільного додатка Dino's Adventure

керівник роботи: Перелигін Борис Вікторович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджений наказом вищого навчального закладу від „18" жовтня 2019 року № 235-с

2. Строк подання студентом роботи: 10. 12. 2019 р.
3. Вихідні дані до роботи:
на основі використання існуючих програмних засобів складання інтерактивних моделей реалізувати та оптимізувати сюжетну побудову створюваного мобільного додатку

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
1. Аналіз операційних систем та вибір ігрового движка для створення ігор
2. Аналіз багатоплатформового ігрового движка Unity
3. Розробка та оптимізація інтерфейсу для ігрового додатка на платформі Android
4. Управління проектом

5. Перелік графічного матеріалу:
1. Презентація роботи

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання:____„28” жовтня 2019 року_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Одержання завдання на виконання магістерської роботи	28.10.2019		
2	Пошук та підбір літератури та інших джерел інформації	31.10.2019		
3	Проведення аналізу предметної області і написання першого розділу пояснювальної записки до магістерської роботи	05.11.2019		
4	Проведення аналізу предметної області і написання другого розділу пояснювальної записки до магістерської роботи	08.11.2019		
5	Проведення аналізу предметної області і написання третього розділу пояснювальної записки до магістерської роботи	14.11.2019		
6	Рубіжна атестація	18-23.11.2019		
7	Проведення аналізу предметної області і написання четвертого розділу пояснювальної записки до магістерської роботи	30.11.2019		
8	Виготовлення презентації	08.12.2019		
9	Друкування пояснювальної записки	09.12.2019		
10	Одержання висновку керівника магістерської роботи	09.12.2019		
11	Проходження нормативного контролю	09.12.2019		
12	Переплетіння пояснювальної записки	09.12.2019		
13	Здача роботи ка кафедру та одержання висновку кафедри про допуск магістерської роботи до захисту	10.12.2019		
14	Перевірка на оригінальність тексту	10.12.2019		
15	Одержання рецензії	10.12.2019		
16	Здача готової магістерської роботи і документів секретарю АК	10.12.2019		
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)			

Студент _____ ___Полєвой Є.В.___
(підпис) (прізвище та ініціали)

Керівник роботи _____ ___Перелигін Б.В.___
(підпис) (прізвище та ініціали)

ПЕРЕДМОВА

Сидячи на просторах інтернету, мені довелося наштовхнутися на статтю "створення гри на платформі Unity". Ознайомившись до статті повністю, я вирішив створити свій, індивідуальний проект, у котрого майже немає аналогів. Я вирішив об’єднати кілька жанрів ігор в один проект, зробити їх з цікавою графікою. Розуміючи, що проект є не зовсім легким і я не зможу його зробити за короткий час, я попросив свого одногрупника, Клименко Микиту Олексійовича, приєднатися до мене. Ми розподілили наші завдання для створення цього проекту, тому я займався:

- розробкою моделей;
- створенням анімації;
- налаштуванням анімації;
- налаштуванням фізики в проекті;
- створенням всіх сцен;
- оптимізацією.

Микита займався:

- розробкою програмного коду;
- прив’язкою скриптів;
- взаємодією між скриптами;
- оптимізацією;
- усуненням програмних помилок.

Спочатку проект розробляли під операційну систему Windows, але обдумавши, вирішили перейти на платформу Android, адже в цю гру не обов’язково грати довгий період часу, а значить що її можна запустити в будь-який вільний час, до того ж гра максимально оптимізована. Ми створили цей проект за досить короткий період часу, а саме за 2 місяці. Надалі є ідеї по створенню ще кілька міні-ігор і запустити її на один з найпопулярніших маркетов – PlayMarket.

РЕФЕРАТ

Кваліфікаційна магістерська робота: 74 с., 52 рис., 1 табл., 1 дод., 25 джерел.

UNITY, 2D МОДЕЛЬ, АНІМАЦІЯ, МОДЕЛЮВАННЯ, СПРАЙТ, СЦЕНА, UI ОБ'ЄКТ, ANDROID.

Мета роботи – створення та експорт на платформу Android мобільного додатка Dino's Adventure та його оптимізація.

Об'єкт роботи – створення 2D моделей, анімації, планування та розробка сцен у середовищі Unity.

Метод дослідження – операційні системи та ігрові движки для створення додатків.

Перший розділ призначений для вибору операційної системи на якій буде базуватися гра і порівняльний аналіз движка для створення гри.

У другому розділі описується платформа движка Unity такі як, інтерфейс, особливості роботи і інші можливості.

У третьому розділі описана практична частина диплому, а саме як були розроблені моделі, як створювалася анімація, як налаштовувалася фізика в грі і як гра була експортована на платформу андроїд.

Четвертий розділ був спрямований на управління проектом. У нього входить планування проекту, управління ризиками проекту і оцінка трудомісткості і швидкості розробки програмного забезпечення.

У результаті проведеної роботи було створено 42 моделі для основної гри таких, як головний персонаж, дерева, бонуси, пастки, платформи, антагоністи та інші, з котрих було також зроблено 12 анімацій. Понад 50 спрайтів було створено для інтерфейсу головного меню, вибір рівня та вибір бажаної гри.

ABSTRACT

Master's qualification work: 74 p., figures 52, tables 1, 1 applications, used literature 25.

UNITY, 2D MODEL, ANIMATION, MODELING, SPRITE, SCENE, UI OBJECT, ANDROID.

The purpose of the work is the creation and export to the Android platform of the Dino's Adventure mobile application and its optimization.

The object of work is the creation of 2D models, animations, planning and scene development in the Unity environment.

The research method is operating systems and game engines for creating applications.

The first section is designed to select the operating system on which the game will be based and a comparative analysis of the engine to create the game.

The second section describes the Unity engine platform such as the interface, features, and other features.

The third section describes the practical part of the diploma, namely how the models were developed, how the animation was created, how physics was configured in the game, and how the game was exported to the android platform.

The fourth section was aimed at project management. It includes project planning, project risk management and assessment of the complexity and speed of software development.

As a result of the work, were created 40 models for the main game such as the main character, trees, bonuses, traps, platforms, antagonists, of which 12 animations were also made, also 50 sprites were created for the main menu interface, level selection and choice of the desired game.

ЗМІСТ

Скорочення та умовні познаки.....8
Вступ.....10
1 Аналіз операційних систем та вибір ігрового движка для створення ігор....12
1.1 Аналіз предметної області.....12
1.2 Огляд існуючих мобільних платформ.....13
1.3 Опис предметної області.....15
1.4 Сучасні ігрові движки.....17
1.5 Переваги багатоплатформового движка.....18
1.6 Огляд існуючих на даний момент ігрових движків.....18
1.6.1 Огляд ігрового движка Corona sdk.....18
1.6.2 Огляд ігрового движка Unreal Engine.....19
1.6.3 Огляд ігрового движка Phaser.....20
1.6.4 Огляд ігрового движка Cocos 2D.....21
1.6.5 Огляд ігрового движка SpriteKit.....22
1.6.6 Огляд ігрового движка CreateJS.....23
1.6.7 Огляд ігрового движка AndEngine.....24
1.6.8 Огляд ігрового движка LibGDX.....25
1.6.9 Огляд ігрового движка Unity.....25
1.7 Вибір движка для ігрового додатка.....27
2 Аналіз багатоплатформового ігрового движка Unity.....28
2.1 Функціональні можливості движка.....28
2.1.1 Можливості графіки у ігровому движку Unity.....28
2.1.2 Можливості звука у ігровому движку Unity.....29
2.1.3 Загальні можливості ігрового движка Unity.....29
2.2 Переваги та недоліки ігрового движка Unity.....30
2.3 Опис інтерфейсу редактора Unity.....31
2.4 Опис використання Accet пакетів у движку Unity.....35
2.5 Оптимізація під роздільну здатність екрана у движку Unity.....36
2.6 Опис використання сторонніх ресурсів у движку Unity.....38
2.6.1 Опис використання спрайтів у движку Unity.....38
2.6.2 Використання шрифтів у движку Unity.....41

3 Розробка та оптимізація інтерфейсу для ігрового додатка на платформі android....42
3.1 Розробка моделей для ігрового додатка....42
3.2 Робота з анімацією у движку Unity....44
3.3 Робота з фізикою, оптимізація та розробка сцен у движку Unity....54
3.4 Експорт ігрового додатка на платформу Android....60
4 Управління проектом....63
4.1 Планування проекту....63
4.2 Управління ризиками проекту....66
4.3 Оцінка трудомісткості і швидкості розробки програмного забезпечення....67
Висновки....70
Перелік джерел посилання....71
Додаток А Презентація роботи мобільного додатку dino’s adventure....73
А.1 Інтерфейс гри Catch....73
А.2 Інтерфейс гри Run....73
А.3 Інтерфейс гри Adventures....74

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API (від Application Programming Interface) – набір методів, констант і структур надаються ОС, додатком або фреймворком для використання у зовнішніх програмних продуктах і використовуваних програмістами при написанні своїх додатків.

Android – операційна система для мобільних платформ (смартфонів, планшетів, електронних книг, ігрових приставок, ноутбуків), телевізорів, і навіть годин. Розробленої на базі Linux компанією Android, Inc., і згодом купленої компанією Google.

GUI (Graphical User Interface) – графічний інтерфейс користувача – забезпечує можливість управління поведінкою системи через візуальні елементи управління – вікна, списки, кнопки, гіперпосилання.

Скрипт – це програма або сценарій, написаний певною мовою, який автоматизує рішення деякої задачі.

TrueTypeFont (TTF) – формат зберігання шрифтів, розроблений компанією Apple в кінці 80-х років. Використовуються в багатьох сучасних операційних системах.

Шейдер (Shader) – це програма для однієї з щаблі графічного конвеєра, яка використовується для завдання остаточних параметрів об’єктів і зображення. Шейдери пишуться на спеціалізованих мовах програмування і компілюються в байткод для виконання на GPU.

Ігровий Ассет (від англ. Game asset) – ресурс, що складається з однотипних даних, що є частиною ігрового контенту і використовуються, оброблювані комп’ютерною грою. Може зберігатися у вигляді файлу

Багатоплатформність (від англ. Cross-platform) – здатність додатки функціонувати на різних апаратних платформах і операційних системах без істотних доробок.

Фізичний движок – це комп’ютерна програма, яка виконує наближене моделювання певних фізичних систем (наприклад динаміка твердого тіла, виявлення зіткнень, динаміка рідин). Широко використовується в комп’ютерній графіці, ігрових додатках і кіно.

Візуалізація (rendering) – процес формування зображення з 2d або 3d моделі або декількох моделей за допомогою програми.

Сцена – сукупність графічних моделей розташованих в певному порядку.

Текстура – растрове цифрове зображення, що відтворює візуальні властивості будь-яких поверхонь або об’єктів.

Спрайт – графічний об’єкт, який можна відобразити на екрані.

Атлас – набір спрайтів або текстур зібраних на одній або декількох великих структурах. Використовується для прискорення доступу до зображень і зменшення кількості перемикань між ними і як наслідок збільшення продуктивності графічної складової програми.

ВСТУП

У сучасних умовах важко уявити собі людину без мобільного телефону, планшетного комп’ютера, смартфона або будь-якого іншого портативного мультимедійного пристрою. Ми звикли до того, що воно завжди під рукою, і це не тільки засіб спілкування, але має і багато корисних функцій, таких як калькулятор, органайзер, конвертер, календар, годинник. Смартфони з нової мобільної ігрової платформою можуть конкурувати з класичними портативними ігровими системами, такими як Nintendo DS або Playstation Portable.

Пристрій смартфона досить просто. В основному він складається з декількох окремих блоків – пам’ять, процесор, який займається організацією обчислень, пам’ять для зберігання даних, радіо-модуль, який, в свою чергу, складається з передавача і приймача і відповідає за зв’язок. Найцікавіше тут – операційна система, встановлена на внутрішній пам’яті. Від операційної системи і її версії залежать і всі основні можливості пристрою. Смартфони, а також персональні комп’ютери, існують з різними операційними системами, види яких будуть обговорюватися далі.

Ігрова індустрія на даний момент найбільш швидко розвивається, особливо це стосується ігрових додатків для мобільних платформ (консолей, смартфонів і планшетів). Зародилася в 70х роках вона зросла на величезну індустрію розваг з оборотами в мільярди доларів. Розробкою ігор зараз займаються як великі компанії (Blizzard Entertainment, Namco, Activision, Bethesda Softworks, Rockstar Games), так і невеликі фірми, спільноти і окремі інді-розробники. Прогрес не стоїть на місці, апаратна частина розвивається, у розробників з’являється більше свободи для творчості і самореалізації. З появою консолей і мобільних телефонів гри перестали бути прерогативою настільних комп’ютерів. Процесори в мобільних пристроях стають швидше, пам’яті більше, зростає потужність мобільних графічних чіпів. Ринок

мобільних ігрових додатків на даний момент є одним з найбільших і розвиваються.

Основними гравцями ринку мобільних платформ є корпорація Google зі своєю операційною системою Android, і Apple просуває мобільну ОС iOS. Обидві операційні системи, ще, будучи молодими, потребували популяризації і розвитку, потребували додатках для реалізації своїх можливостей, для чого компанія Google і Apple створили свої магазини додатків (Google play і Apple store). Магазин додатків дозволяв користувачам знаходити, завантажувати та встановлювати програми сторонніх розробників, які в свою чергу пропонували свої додатки за гроші або на безкоштовній основі. Розробка мобільних додатків стала надприбутковим бізнесом при мінімальних вкладеннях. Особливо швидко розвивався ринок саме мобільних ігор, величезна кількість ігор було перенесено на мобільні платформи з настільних комп’ютерів.

Згідно зі статистикою, більше половини мобільних пристроїв працюють під управлінням операційної системи Android. Очевидно, що розробка ігор під цю операційну систему зараз як ніколи актуальна. Ідея створення гри як дипломного проекту була продиктована моїм бажанням спробувати свої сили в розробці ігор (моя робота не пов’язана з цією областю), використовувати навички в розробці і використання інформаційних систем, отримані в процесі навчання у вузі.

На початку даної роботи ми розглянемо цілі і вимоги, що пред’являються до нашого ігрового додатку, і способи їх досягнення. Далі проведемо порівняльний аналіз і огляд ігрових движків с подальшим вибором кращого, і задовольняє нашим вимогам. У третьому розділі ми розробимо і розберемо структуру нашого застосування, розробимо меню, призначений для користувача інтерфейс, попутно вивчаючи можливості інструментарію і засобів розробки входять до складу обраного нами движка.

1 АНАЛІЗ ОПЕРАЦІЙНИХ СИСТЕМ ТА ВИБІР ІГРОВОГО ДВИЖКА ДЛЯ СТВОРЕННЯ ІГОР

1.1 Аналіз предметної області

В даний час комп’ютерні технології стали активно застосовуватися в різних сферах нашого життя. Створення різних простих і складних комп’ютерних програм для різних областей знань постійно розвивається. Залежно від віку дитини комп’ютерна програма може виступати в якості опонента по грі, бути оповідачем, репетитором, екзаменатором. Вже існують комп’ютерні програми, спрямовані на розвиток різних психічних функцій дітей, таких, як візуальне і слухове сприйняття, увагу, пам’ять, словесні і логічні міркування і т.д, які можуть бути успішно використані дітьми старшого дошкільного і молодшого шкільного віку. Тепер на зміну настільним комп’ютерам приходять мобільні – планшетні ПК, смартфони, що дозволяє більш зручно працювати з усією інформацією.

Гра – це пізнавальна діяльність, вона є свого роду практичною формою мислення про природу навколишнього дитини соціальної реальності. Через особливості гри, дитина в грі вперше долучається до абстрактного мислення.

Комп’ютерні ігри – це новий вид розвиваючого навчання. Комп’ютерні технології відкривають нові можливості для використання педагогічних методів:

- використання матеріалу різної складності. Індивідуально дитині завжди можна запропонувати саме те, що в даний момент відповідає його можливостям і цілям навчання;
- для того щоб "бачити" проблеми в розвитку дитини, які важко виявити в традиційному навчанні;
- для того щоб сформувати розуміння дитиною процесу отримання власних навичок;

- складне програмне забезпечення є надзвичайно простим у використанні;
- багато методів успішно використовувалися раніше, тепер поставлені на комп'ютерну базу і отримали як би друге розвиток. З точки зору фахівця, це можливість поглянути на свою роботу під новим кутом зору, переосмислити методичні прийоми, збагатити знання і навички, якими він володіє;
- робота на комп'ютері створює для дитини більш комфортні умови для успішного виконання вправ;
- комп'ютерні технології забезпечують захоплюючі заняття для дитини в формі експериментування, моделювання, порівняння;
- дитина навчиться правильно говорити, намагатися виправити ту помилку, що він побачив, шукаючи методи самоконтролю, зосередивши увагу на привабливою графіку;
- діти менше втомлюються, більше здатні працювати;
- дивлячись на екран, дитина сама бачить результат своєї роботи.

Таким чином, використання комп'ютерної програми підвищує мотивацію використання не тільки стратегії гри, на якій базується програма, а й того, що дитина отримує схвалення, похвалу, не тільки дорослих, а й комп'ютера.

1.2 Огляд існуючих мобільних платформ

Наявність операційної системи (ОС) – головна особливість, яка відрізняє смартфон від звичайного мобільного телефону. При виборі конкретної моделі телефону або пристрою, операційна система часто є визначальним фактором [1][1)].

[1)1)] [1] Програмні платформи сучасних смартфонів. URL : http://ukrefs.com.ua/print:page,1,-155256-Programmnye-platformy-sovre mennyh-smartfonov.html. (дата звернення 1.11.2019).

Найбільш поширені операційні системи для смартфонів і платформ:

- Symbian OS – ОС займала більшу частину ринку смартфонів до кінця 2010 р. На початку 2010 року на базі ОС залишається тільки 1 Платформа: Series 60, яка використовується в основному в пристроях Nokia, а також деякі моделях Samsung;
- BlackBerry OS (RIM) – система широко використовується в пристроях в першу чергу в Сполучених Штатах, так як спецслужби деяких країн не зацікавлені у використанні смартфонів в країні через те, що всі вхідні і вихідні дані зашифровані з використанням алгоритму шифрування AES;
- Windows Mobile і Windows CE – компактна операційна система Microsoft, випущені з 1996 року і займала найбільший сегмент ринку ОС для смартфонів до 2010 року, в даний час проходить поетапну відмову від підтримки і розвитку;
- Windows Phone 7 – розробка від Microsoft, радикально відрізняється від Windows Mobile;
- Palm OS – одна з популярних платформ, хоча даний час мобільні телефони на базі Palm OS малораспространени. Останній смартфон під управлінням операційної системи була випущений в кінці 2007 року (Palm Centro);
- Linux – широкого поширення ця операційна система на мобільних пристроях не отримала, проте її розвиток традиційно вважається перспективним напрямком. Смартфони на базі Linux поширюються головним чином в Азії;
- Bada – новітня мобільна платформа, розроблена компанією Samsung. Першим телефоном на новій платформі став S8500 Wave;
- Android – портативна (мережева) операційна система для смартфонів, планшетих ПК, електронні книг, цифрових плеєрів, годин і нетбуків на базі ядра Linux. Спочатку розроблена Android Inc., яку

потім купив Google. Згодом Google ініціювала створення альянсу Open Handset Alliance (OHA), який зараз займається підтримкою і подальшим розвитком платформи. Android дозволяє створювати додатки на основі Java, який управляють пристроєм через розроблені Google бібліотеки. Android Native Development Kit дозволяє системі використовувати бібліотеки і компоненти додатків, написаних на C та іншими мовами;

– ОС IOS (до 24 червня 2010 року – iPhone OS) – це мобільна операційна система, розроблена і виготовлена американською компанією Apple. Вона була випущена в 2007 році; спочатку – для iPhone і IPod Touch, а пізніше – для таких пристроїв, як IPad і Apple TV. На відміну від Windows Phone і Google Android, доступна тільки для пристроїв, вироблених Apple; [2][1)]

1.3 Опис предметної області

Операційна система Android заснована на платформі Linux для мобільних пристроїв, розроблених Open Handset Alliance (OHA), ініційованої Google. Вона дозволяє створювати додатки на основі Java, які керують пристроєм через бібліотеки, розроблені Google. Крім того, можна писати програми на C та іншими мовами програмування за допомогою Android Native Development Kit.1.5 (Cupcake), випущеної 30 квітня 2009 г. Серед основних поліпшень з’явилася підтримка запису і перегляду відео в режимі камери; підтримка Bluetooth A2DP; можливість автоматичного підключення до Bluetooth гарнітури.

Перший пристрій, яке працювало під управлінням Android, став розроблений HTC смартфон TMobile G1, який був запущений 23 вересня

---

1)1) [2] Яку вибрати платформу для створення мобільної гри | Каталог статей. URL : https://catalogueofarticles.com/uk/tehnologiyi/jaku-vibrati-platformudlja-stvorennja-mobilnoi-gri/ (дата звернення 02.11.2019).

2008 року Незабаром пішли численні анонси інших виробників смартфонів, які планують випустити пристрої з Android.

Є кілька основних переваг Google, які відрізняють пристроїв на базі платформи Android від аналогічних продуктів:

- відкритість – Android дозволяє отримати доступ до основних функцій мобільного пристрою, використовуючи стандартний API виклик;
- руйнування кордонів – можна об'єднувати інформацію з Інтернету з даними телефону, наприклад, контактну інформацію або географічних даних про місцезнаходження, щоб отримати нові можливості;
- рівність додатків – для Android немає ніякої різниці між основними телефонними додатками і програмним забезпеченням сторонніх виробників – ви можете навіть змінити програму для набору номера або заставку;
- швидко і легко розробляти програми – в SDK є все необхідне для створення і запуску Android-додатків, в тому числі імітатор справжнього приладу і передові інструменти налагодження.

Гнучкість Android має ціну: компанії, вважаючи за краще розробляти свої власні інтерфейси, постійно прагнуть до випусків нових версій ОС. Пристрої, що випустили кілька місяців тому, стають застарілими, оскільки оператори і виробники не хочуть створювати оновлення програмного забезпечення, так що користувачі можуть використовувати нові функції Android. Наприклад, багато експертів відзначають, що платформа заснована на Java, тому переваги і можливості операційної системи Linux на Android не використовуються повною мірою. Вона також не використовує в платформі будь-які з популярних графічних інструментів (Toolkit) і бібліотек (наприклад, Ot або GTK), що робить малоймовірною появу великої кількості додатків, перенесених на Linux з повною версією для домашнього

комп'ютера на мобільну платформу через відсутність загального сервера і бібліотеці зображень.

Крім того, повідомлялося, що Google буде на свій розсуд видаляти додаток на телефонах користувача, якщо порушуються умови їх використання.

Аналітики і експерти пророкують для IT-ринку Google Android хороші комерційні перспективи, в принципі, для продуктів на базі програмного забезпечення з відкритим вихідним кодом, що вже не є сенсацією. Вони поступово займають IT простір, витісняючи з нього визнаних лідерів, створюючи конкуренцію, що само по собі може тільки позитивно вплинути на відновлення ринку.

Велика частина коду ліцензується відповідно до ліцензії Apache 2, відкрита і необмежена ліцензія дозволяє вільне використання вихідного коду для створення своїх власних систем.

1.4 Сучасні ігрові движки

Ігровий движок являє собою набір підсистем, контролюючих певні частини гри. Більшість з ігрових движків мають наступні підсистеми:

– графічна підсистема;
– підсистема ввода-вивода;
– звукова підсистема;
– мережева підсистема;
– ядро.

Використання вбудованих в движок візуального редактора, готових модулів рендеринга, анімації спрайтів і обробки зіткнень спрощує процес розробки додатків.

Багатоплатформовий движок зробить можливим перенесення вашого застосування на іншу апаратну платформу, без будь-яких серйозних змін в

коді.

Передбачена в складі движка інтегроване середовище розробки, котра дозволить зробити код більш читабельним, і організованим [3],[4][1)].

1.5 Переваги багатоплатформового движка

Плюси використання багатоплатформових мобільних додатків:

- можливість використовувати один і той же код в інших проектах і на інших операційних системах знову і знову;
- доступ розробників до плагінів і модулів, можливість з вбудовування в інші сервіси, інструменти;
- компаніям тепер немає сенсу наймати розробників під різні платформи;
- можливість інтеграції в хмарні сервіси;
- простота розгортання прискорює процес розробки мобільного застосування;
- додатки вимагають, мобільного реалізації охоплюють такі категорії як фото, відео, новини, бізнес, ігри, книги та розваги.

Найбільш підходящим для розробки будь-якого сучасного додатки, є багатоплатформові рішення, які забезпечують високу швидкість роботи на широкому колі популярних мобільних платформ і зручність розробки.

1.6 Огляд існуючих на даний момент ігрових движків

1.6.1 Огляд ігрового движка Corona sdk

Corona SDK являє собою багатоплатформовий движок для створення ігор під iOS і Android. API дозволяє робити 2D-ігри, а також допоможе вам

[1)1)] [3] List of game engines and frameworks. URL: https://en.wikipedia.org/wiki/List_of-_game_engines (дата звернення 08.11.2019).

[4] DevGam сравнение игровых движков URL : http://devgam.com/sravne-nie igrovyx-dvizhkov-kakoj-vybrat (дата звернення 08.11.2019).

створювати складні функції, використовуючи API на основі Lua. Крім того, ви можете монетизувати Corona SDK за допомогою Corona Ads (рис. 1).

Рисунок 1 – Логотип ігрового движка Corona SDK

Corona включає в себе такі цікаві інструменти, як Corona Editor, Sublime Text plugin і багато інших.Composer GUI, доступний на OS X, надасть вам графічне середовище, в якій ви зможете створювати рівні для ігор і бачити, як об’єкти взаємодіють один з одним, використовуючи Corona’s physics engine. Corona SDK надає інструментарій для створення багатоплатформових ігрових додатків. Є можливості для монетизації за допомогою Corona Ads [5][1)].

1.6.2 Огляд ігрового движка Unreal Engine

UE4 дозволяє створювати проекти високої якості силами невеликої групи розробників за рахунок потужного інструментарію (зокрема, візуального скріптованія допомогою Blueprint), відкритого вихідного коду і постійного зростання ком’юніті. Є безкоштовна версія Unreal Engine, перехід уа Pro-версію обійдеться в кругленьку суму (рис. 2).

Візуалізація в редакторі чудова. Просто очі розбігаються від достатку опцій рендеринга (пов’язаних, наприклад, з освітленням або зі складністю

[1)1)] [5] Corona SDK – Википедия URL https://ru.wikipedia.org/wiki/Corona_SDK(дата звернення 10.11.2019).

шейдеров). Тут ви знайдете масу ультрасучасних шейдеров, які також поставляються разом з движком.

Рисунок 2 – Логотип ігрового движка Unreal Engine

Unreal пропонує найкращий механізм рендеринга на ринку. Можна створювати дивовижно красиві сцени. Unreal Engine, один з найбільш просунутих ігрових движків. Може бути використаний абсолютно безкоштовно, до того моменту коли ваш дохід перевищить 3000 $ [6][1)].

1.6.3 Огляд ігрового движка Phaser

Phaser (рис. 3) – це фреймворк для розробки десктопних і мобільних HTML5-ігор, що базується на бібліотеці PIXI.js. Підтримує рендеринг в Canvas і WebGL, анімовані спрайт, частинки, аудіо, різні способи введення і фізику об’єктів.

[1)][1)] [6] Unreal Engine – Википедия. URL https://ru.wikipedia.org/wiki/ Unreal_Engine. (дата звернення 10.11.2019).

Рисунок 3 – Логотип ігрового движка Phaser

Phaser поширюється з відкритим вихідним кодом за ліцензією MIT.. Це означає, що ви можете використовувати код без обмежень, але зі збереженням повідомлень про авторське право в копіях ПО, тобто в тексті вашої ліцензії необхідно буде додати вказівку авторських прав на цей фреймворк.

Він добре підтримується Річардом Деві і з'явилися навколо нього спільнотою. Phaser HTML5 движок, який має відкритий вихідний код. Хороший варіант для тих, хто створює web ігри [7][1)].

1.6.4 Огляд ігрового движка Cocos 2D

Cocos2d – це open-source фреймворк, сумісний з Swift і Objective-C. Він має підтримку iOS і OS X, а також Android через плагін SpriteBuilder Android (за умови, що ваш код написаний на Objective-C). Підтримка Swift для Android знаходиться в стадії розробки.

Проекти в Cocos2d створюються через SpriteBuilder – графічне середовище розробки, яку можна використовувати для швидкого створення прототипів та ігор (рис. 4).

Управління сценою здійснюється через клас CCDirector, який може використовувати численні переходи з класом CCTransition. Він пропонує анімацію за допомогою класу і дій, таких як переміщати, масштабувати і обертати CCAnimation з його CCAction-класу.

Cocos2d має підтримку систем частинок з класом CCParticleSystem і карт плитки з класом CCTiledMap. Також в співтоваристві Cocos2d є кілька незалежних редакторів, призначених для редагування спрайтів, частинок, шрифтів і тайлових карт. Можна також згадати редактори світу: CocosBuilder

---

[1)] [7] Phaser (game framework) – Wikipedia URL: https://en.wikipedia.org/wiki/Phaser_(game_framework) . (дата звернення 10.11.2019).

і CocoStudio [8] [2)].

Рисунок 4 – Логотип ігрового движка Cocos2d

1.6.5 Огляд ігрового движка SpriteKit

SpriteKit – 2Dдвижок, оптимізований під створення ігор для Apple пристроїв (рис. 5). В його основі лежить фізичний движок Box 2D[9] [1)].

[2)2)] [8] Cocos2d – Википедия URL: https://ru.wikipedia.org/wiki/Cocos2d (дата звернен-ня 11.11.2019).

[1)1)] [9] Создание игры на движке Sprite Kit (Часть 1) / Хабр URL: https://habr.com/ru/-post/225517/. (дата звернення 11.11.2019).

Рисунок 5 – Логотип ігрового движка SpriteKit

Оскільки розробники створювали його заточеним спеціально для пристроїв Apple, він істотно виграє в швидкості у інших движків. Надає всі основні функції для створення графіки в іграх: підтримку OpenGL-ES-шейдеров, ефектів освітлення, анімації, перевірки зіткнень, отрисовку тексту, відео і так далі.

1.6.6 Огляд ігрового движка CreateJS

CreateJS (рис. 6) – це збірка різних бібліотек для створення ігор з відкритим вихідним кодом:

- EaselJS,
- TweenJS,
- SoundJS,
- PreloadJS.

EaselJS – це бібліотека, яка заснована на Canvas API.

TweenJS – це анімація в Canvas. І вона відмінно справляється зі своїм завданням, роблячи за вас і спрощуючи багато речей.

SoundJS, як ви вже здогадалися, призначена для роботи з аудіо.

PreloadJS допомагає керувати і координувати завантаження даних.

Ці бібліотеки можуть працювати як разом, так і окремо. Кожен модуль відповідає за свою частину роботи і не перетинається з іншими, тому при необхідності можна замінити на аналогічний за функціональністю від інших розробників.

Але для тих, хто має намір використовувати весь набір без змін, передбачена можливість їх зв’язування для спрощення роботи і написання коду [10][1)].

[1)] [10] CreateJS – Wikipedia URL: https://en.wikipedia.org/wiki/CreateJS (дата звернення 11.11.2019).

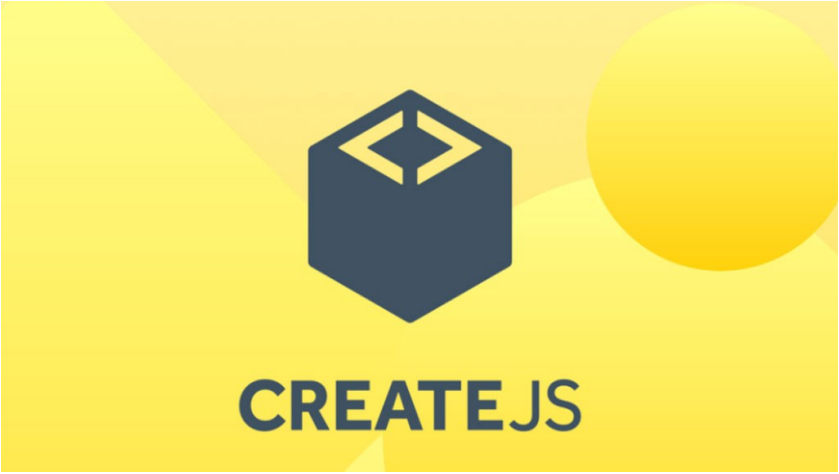


Рисунок 6 – Логотип ігрового движка CreateJS

1.6.7 Огляд ігрового движка AndEngine

AndEngine є одним з найбільш відомих безкоштовних движків з відкритим вихідним кодом для написання 2D-ігор під Android [11][1)]. AndEngine, як і багато інших платформи, використовує OpenGL для рендеринга і Box2D для фізики через розширення (рис. 7).

Рисунок 7 – Логотип ігрового движка AndEngine

[1)1)] [11] AndEngine – Wikipedia URL: https://en.wikipedia.org/wiki/AndEngine (дата звернення 11.11.2019).

1.6.8 Огляд ігрового движка LibGDX

LibGDX (рис. 8) – це багатоплатформовий фреймворк, який написаний на Java і працює з OpenGL. Пропонує підтримку декількох платформ для публікації гри [12][1)].

Рисунок 8 – Логотип ігрового движка LibGDX

Ingress (попередник Pokémon GO) був розроблений з використанням libGDX. LibGDX – фреймворк для створення ігор і додатків, написаний на Java з використанням C і C ++. Він дозволяє писати Кросплатформені ігри та програми використовуючи один код Підтримка спільноти також чудова, завдяки цьому ви можете отримати відмінну документацію для роботи.

1.6.9 Огляд ігрового движка Unity

Unity вважається одним з кращих ігрових движків в галузі. Є безкоштовна версія, яка може бути використана для створення 2D і 3D-ігор. Вона охоплює 24 платформи: мобільні пристрої, VR, десктоп, консолі і веб-платформи (рис. 9).

Проект в Unity ділиться на сцени (рівні) – окремі файли, що містять

---

1)[1)] [12] LibGDX – Википедия. URL: https://ru.wikipedia.org/wiki/LibGDX. (дата звер-нення 11.11.2019).

свої ігрові світи з набором об’єктів, сценаріїв і налаштувань. [13][1)].

Рисунок 9 – Логотип ігрового движка Unity

Для невеликих інді-компаній або окремих розробників принципово щоб інструментарій для створення ігор коштував дешево, не мав зайвого функціоналу, був простий в освоєнні. Для них є важливими такі аспекти як:

- Багатоплатформність для забезпечення доступності додатки на якомога більшій кількості платформ, і переносимість коду без істотних змін;
- движок повинен бути розширюємо, бути модульним;
- можливість освоїти движок в максимально короткі терміни;
- розробка ігрового додатки не повинно вимагати істотних фінансових вливань;
- не висока кваліфікація розробників не повинна бути проблемою для створення програми.

1)1) [13] Unity (игровой движок) – Википедия. URL : https://ru.wikipedia.org/wiki/Unity- (игровой_движок). (дата звернення 11.11.2019).

1.7 Вибір движка для ігрового додатка

Головна ідея цього проекту – багатоплатформність, вибір на даний момент серед таких продуктів невеликий. Ось кілька найпопулярніших, представлених на ринку багатоплатформових движків: Unreal Development Kit, Unity, Godot Engine, Cocos2D, MonoGame. Всі ці движки мають дуже хороші технічні показники і підтримують більшість мобільних платформ. Свій вибір хотілося б зробити виходячи не тільки з кількості підтримуваних платформ, але і легкості перенесення коду на різні платформи, продуктивність і інше. Перш ніж зробити остаточний вибір, наведемо список факторів, на які слід звернути увагу при виборі движка для розробки гри:

- Кількість підтримуваних платформ і легкість складання проекту під них;
- мову програмування;
- витрати на навчання розробників;
- вартість ліцензії або величина відрахувань розробникам движка;
- продуктивність, як на сучасних, так і на старих слабких платформах і пристроях.

Орієнтуючись на всі перераховані вище фактори, і на основі аналізу наведеного в попередніх розділах, можна з упевненістю сказати, що найбільш підходящим для нашого проекту буде Unity. Unity володіємо всіма тими якостями, які так нам потрібні: це і продуктивність в 3D і особливо в 2D, і легка переносимість проекту на різні платформи, і середовище розробки "з коробки", і вбудований редактор спрайтів і анімації, і багато іншого.

2 АНАЛІЗ БАГАТОПЛАТФОРМОВОГО ІГРОВОГО ДВИЖКА UNITY

2.1 Функціональні можливості движка

Основним критерієм вибору того чи іншого багатоплатформового ігрового движка є ставлення його функціональності до вартості. У даній роботі нам необхідно створити двовимірний багатоплатформовий ігровий додаток. Нижче будуть наведені основні критерії, за якими можна охарактеризувати Unity як сучасний ігровий движок з багатим потенціалом і широкими можливостями [14][1)].

2.1.1 Можливості графіки у ігровому движку Unity

Графіка у середовищі Unity має такі здібності, як:

– система частинок. Спосіб представлення об’єктів, що не мають чітких геометричних кордонів (хмари, туманності, дим, пар, рідини і т.д.);
– ефект рендеру в текстуру, тобто можливість приміщення зображення з камери нема на екран, а в текстуру. Потрібно для створення деяких ефектів;
– автоматична нарізка і анімація спрайтів. вбудований редактор спрайтів здатний автоматично нарізати растр з спрайтами на кадри і створювати з них анімацію. Прив’язувати події до певних ключовим кадрам анімації;
– пакувальник спрайтів, використання різноманітних видів стиснення текстур і упаковка використовуваних кадрів спрайтів в один ресурс;
– 2D фізика. Вбудована бібліотека Box2D, і підтримка NVIDIA PhysX (фізика реалізована за рахунок використання в розрахунках GPU);

---

1)1) [14] Руководство Unity URL : https://docs.unity3d.com/ru/current/Manual/index.html (дата звернення 12.11.2019).

– альфавирез, обрізання частини зображення по масці, яку розміщують в альфакомпоненте текстури;
– виявлення зіткнень;
– Dynamic Batching, алгоритм оптимізації рендеринга дозволяє домогтися збільшення продуктивності.

2.1.2 Можливості звука у ігровому движку Unity

Звук у середовищі Unity має такі здібності, як:
– власні аудіо плагіни, можливість використання сторонніх аудіо плагінів;
– виклик скриптів з анімації. Створення ключових кадрів в редакторі анімації і прив'язки до них скриптів;
– зміни настрою звуку [16][1)].

2.1.3 Загальні можливості ігрового движка Unity

У ігровому движку Unity також є такі здібності, як:
– функції з передовою системою пошуку шляху;
– реалістична анімація;
– розміщення одним клацанням;
– оптимізована графіка;
– конвеєр імпорту;
– 64-розрядний розширюваний редактор.

2.2 Переваги та недоліки ігрового движка Unity

[1)] [15] Unity - Руководство: Audio Source. URL https://docs.unity3d.com/ru/current/Manual/class-AudioSource.html. (дата звернення 12.11.2019).

У порівнянні з іншими двигунами, Unity має такі переваги:

– простота використання і продуктивність;
– величезний число підтримуваних платформ: Windows, iOS, Android;
– можливість запуску проекту в браузері як HTML5 додатки;
– гарна продуктивність як в невеликих іграх на слабких мобільних платформах, так і в складних великих проектах на high end консолях;
– підтримка як 2D так і 3D режимів без особливих зусиль;
– потужна вбудована анімаційна система Mechanim;
– мова програмування C # або JavaScript;
– велике співтовариство, де можна, можливо отримати підтримку і відповіді на цікавлять питання по Unity;
– вбудований візуальний редактор, сильно спрощує і прискорює процес створення програми;
– assets store, магазин для розробників, що відкриває доступ до величезного числа моделей, текстур, скриптів і доповнень для Unity;
– можливість використання абсолютно безкоштовною середовища розробки MonoDevelop йде в комплекті з Unity, якої зовнішньої IDE (наприклад, Microsoft Visual Studio або IntelliJ IDEA);
– усі функції Unity абсолютно безкоштовні до досягнення доходу з продажу в 100 тисяч доларів.

До недоліків Unity можна віднести наступне:

– ісходний код Unity закритий і доступний тільки в разі покупки Enterprise ліцензії. У разі будь-яких проблем, доведеться чекати поновлення і виправлень, що іноді займає досить багато часу;
– відсутність такого звичного поняття як "game loop" або ігрового циклу. Немає єдиної точки входу, як в інших двигунах. Кожен ігровий об’єкт може мати скрипт або кілька скриптів, свій набір подій і свій ігровий цикл;

– не звичний і місцями складний в освоєнні візуальний редактор, який може виявитися проблемою для початківців розробників.

2.3 Опис інтерфейсу редактора Unity

Редактор складається з декількох основних частин, які можна перетягувати, групувати і розташовувати, так як нам зручно (рис. 10).

Вікно Project, дозволяє нам управляти, добавляти або видаляти [16][1)] об’єкти всього проекту (рис. 11).

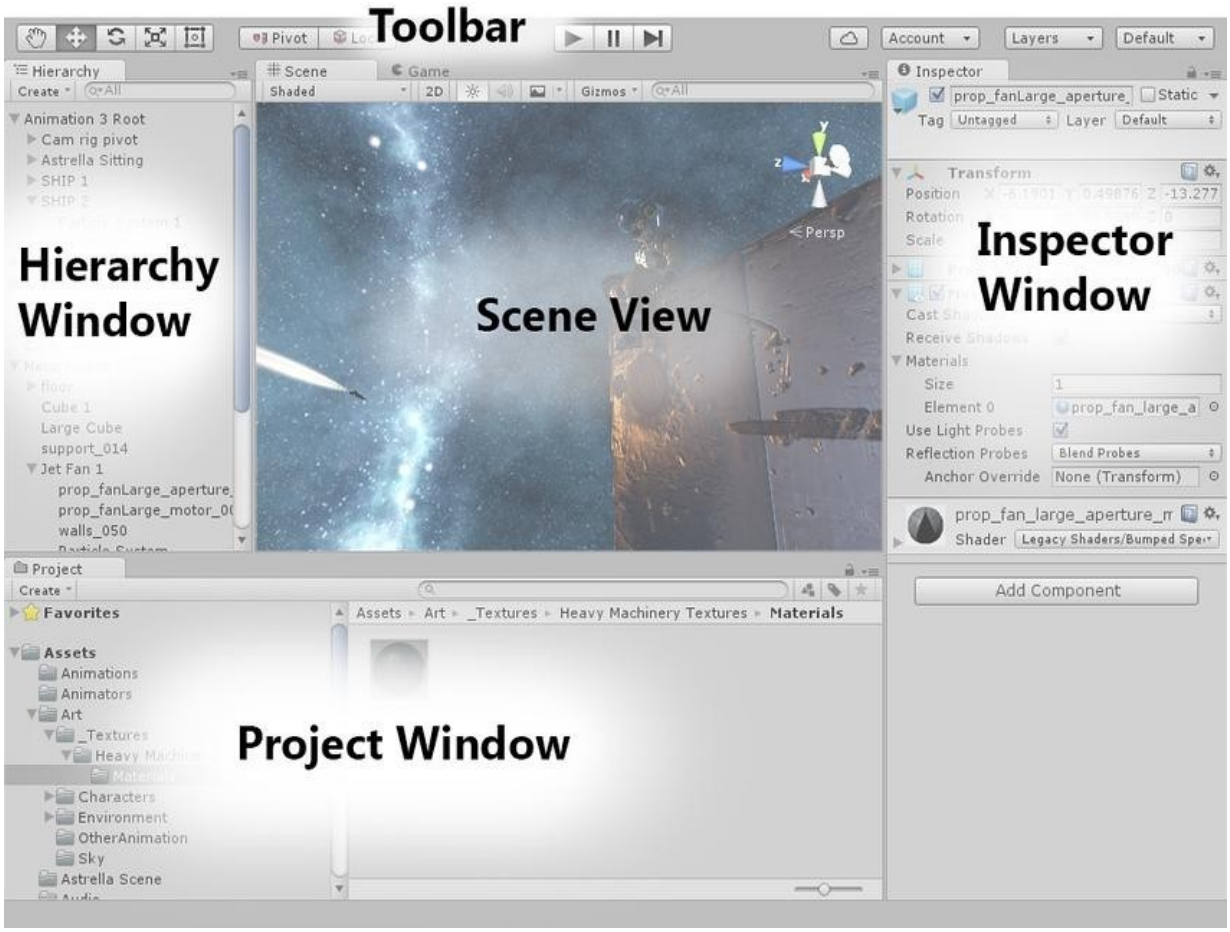


Рисунок 10 – Інтерфейс ігрового движка Unity

[1)] [16] Unity - Руководство: Изучение интерфейса URL https://docs.unity3d.com/ru/current/Manual/LearningtheInterface.html. (дата звернення 12.11.2019).

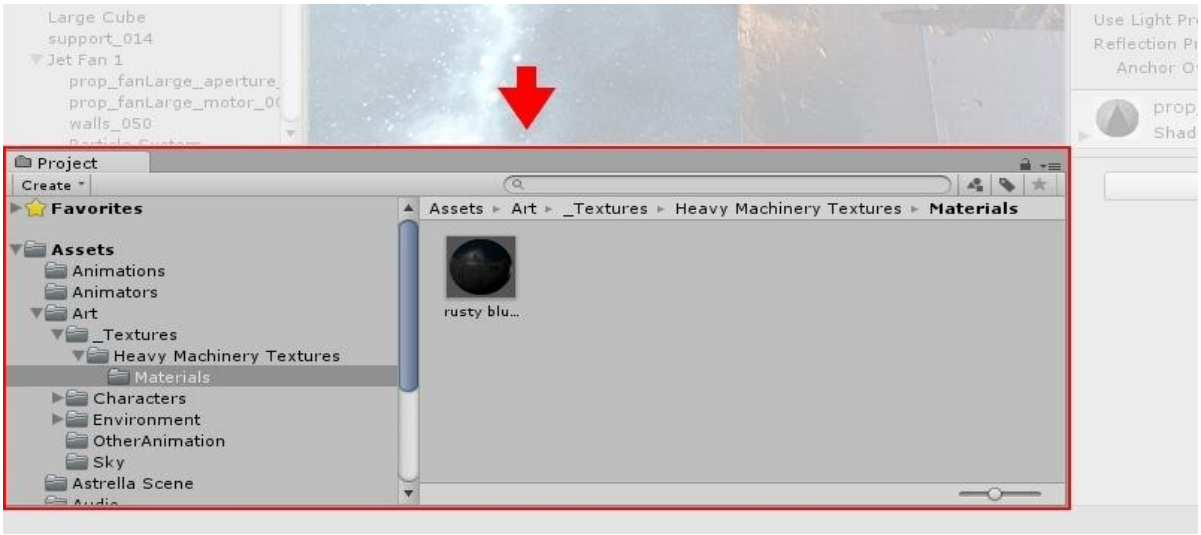


Рисунок 11 – Інтерфейс вікна «Project» у движку Unity

Вікно Scene, дозволяє нам управляти, редагувати сцену або кілька сцен проекту, тобто вікно, де будуть відображатися об’єкти нашої гри (рис. 12). Вікно має можливість перемикатися в 2D або 3D режими, в залежності від типу нашого проекту, вмикати і вимикати освітлення. Дозволяє вибирати, переміщати і змінювати позицію об’єктів знаходяться на сцені (наприклад: джерела світла, персонажа, камери та інші), тимчасово вимикати об’єкти.

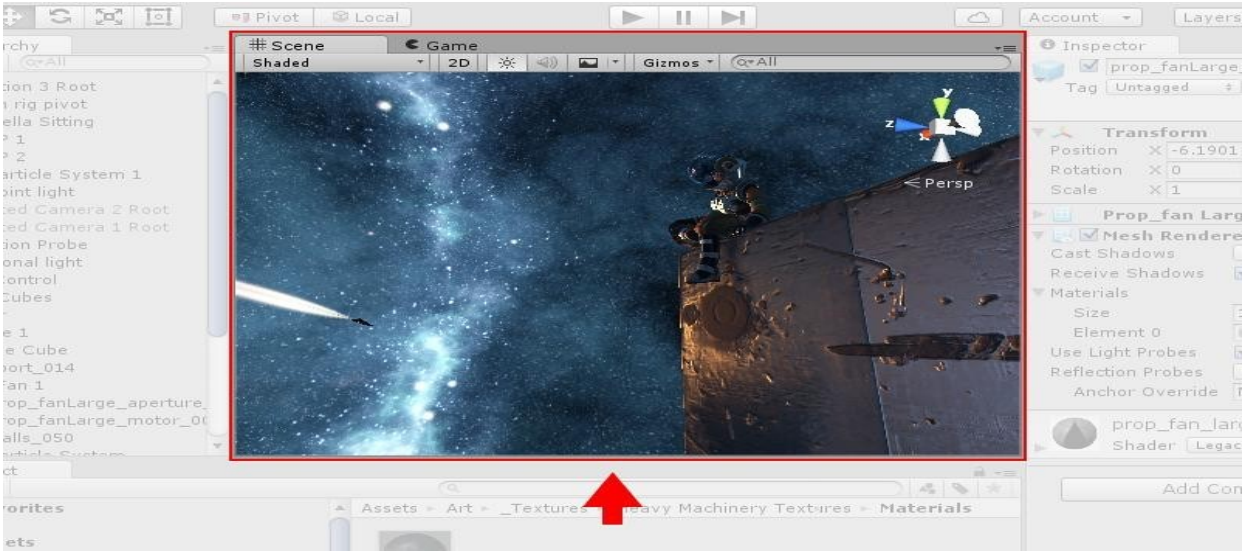


Рисунок 12 – Інтерфейс вікна «Scene» у движку Unity

Вікно Hierarchy відображає ієрархію всіх об’єктів знаходяться на сцені. Кожен елемент сцени має запис в цьому вікні. вікно Hierarchy відображає структуру сцени (рис. 13) і як об’єкти прикріплені один до одного, воно нерозривно пов’язане з вікном Scene.

Вікно Inspector дає нам можливість переглядати і змінювати всі можливі параметри обраних об’єктів. Так як параметри різних об’єктів відрізняються, то розташування і вміст вікна інспектора може змінюватися в залежності від обраного об’єкта.

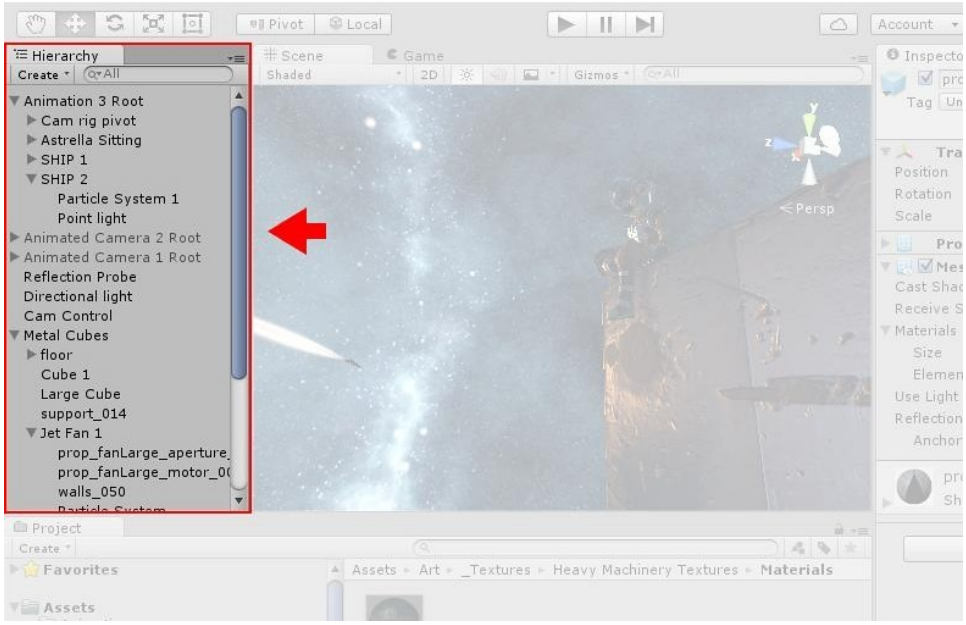


Рисунок 13 – Інтерфейс вікна «Hierarchy» у движку Unity

Інспектор (рис. 14) використовується для перегляду та зміни параметрів і налаштувань майже все, що є в редакторі Unity, включаючи ігрові об’єкти, матеріали, спрайт, Ассет і так далі.

Якщо до об’єкта доданий скрипт, то Інспектор (рис 15) буде відображати всі публічні змінні і посилання на інші об’єкти, присутні в ньому. Вміст цих змінних можна змінювати прямо в цьому вікні, не залазячи в скрипт і не змінюючи код безпосередньо.

Toolbar надає доступ до найбільш часто використовуваних функцій ре-

дактора (рис. 16). Справа знаходяться кнопки доступу до хмарних сервісів і аккаунту Unity. Layers відповідає за видимість шарів на сцені, дозволяє включати і вимикати їх у процесі налагодження. У середній частині панелі знаходяться кнопки управління роботою додатка, що дозволяють запускати, ставити на паузу і покроково його виконувати. Ліва частина містить стандартні інструменти управління сценою і об’єктами на ній.

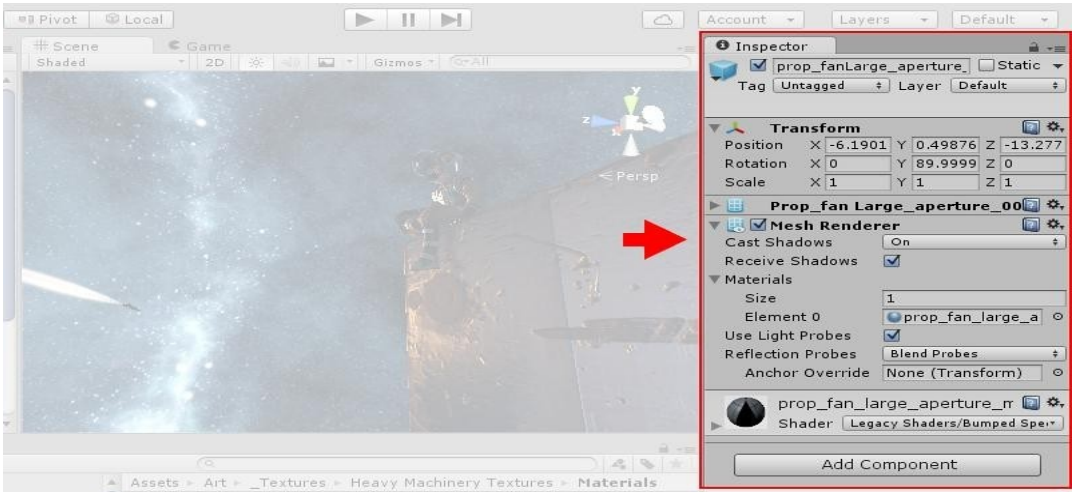


Рисунок 14 – Інтерфейс вікна «Inspector» у движку Unity

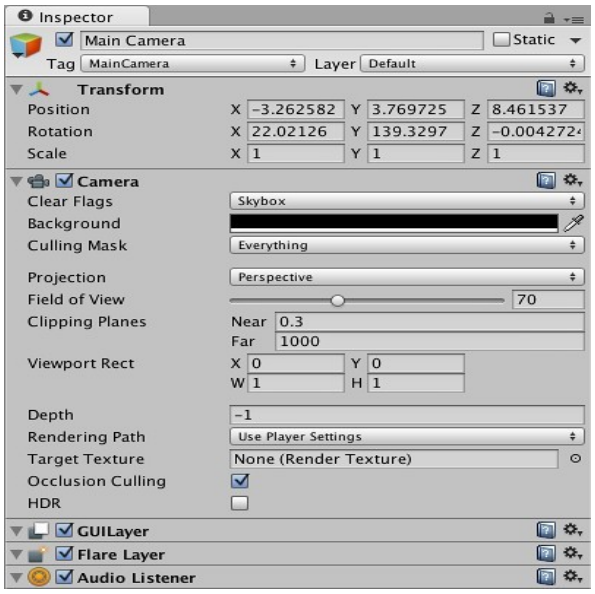


Рисунок 15 – Інтерфейс вікна «Inspector Camera» у движку Unity

Рисунок 16 – Інтерфейс вікна «Toolbar» у движку Unity

2.4 Опис використання Accet пакетів у движку Unity

Разом з Unity при установці ми отримуємо стандартний набір Ассет, що складається з декількох різних пакетів, які можуть знадобитися починаючому розробнику. Для їх установки потрібно або вказати і вибрати їх при створенні проекту, або вже після створення проекту в редакторі зайти в меню Assets і вибрати Import assets package. У вікні вибрати необхідні ресурси, і вони будуть імпортовані в проект (рис. 17).

Так само є можливість імпорту ресурсів зроблених іншими людьми, процес абсолютно аналогічний попередньому, за винятком того що потрібно вказати шлях до пакету, який ви хочете імпортувати [17][1)].

Для того щоб полегшити і прискорити процес створення ігор і дати можливість людям отримувати гроші за свої розробки і залучити нових користувачів движка, розробники Unity створили магазин Ассет (Unity asset store), який являє собою інформаційну систему, яка надає доступ до розробок інших користувачів Unity. Кожен користувач движка Unity має можливість помістити в цей магазин свої розробки. Це може бути і повноцінна гра, і набір скриптів полегшує процес розробки, текстури або моделі, системи частинок, шейдери, ігровий арт та багато іншого. Все це поширюється в вигляді пакетів і може бути як платним, так і безкоштовним. Магазин вже інтегрований в редактор Unity, ми просто вибираєте вкладку Asset store, знаходите те що вас цікавить, завантажуєте і відразу ж імпортуєте в проект.

---

1)[1)] [17] Движок Unity – особенности. URL : http://gamesmaker.ru/3d-gameengines/unity- 3d/osobennostibesplatnogo-krossplatformennogo-dvizhka-unity3d (дата звернення 12.11.2019).

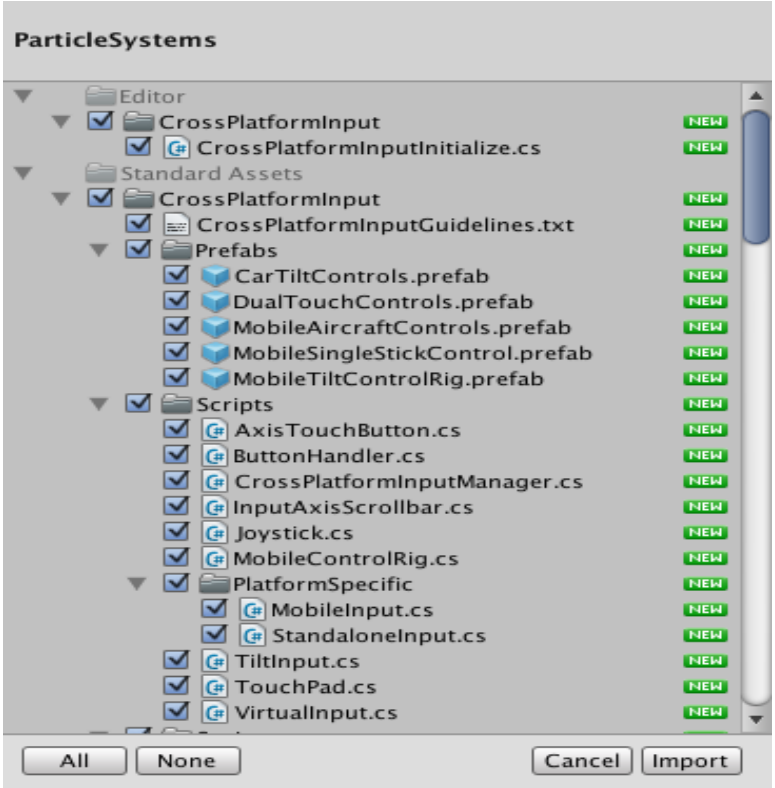


Рисунок 17 – Інтерфейс вікна «Assets Import» у движку Unity

2.5 Оптимізація під роздільну здатність екрана у движку Unity

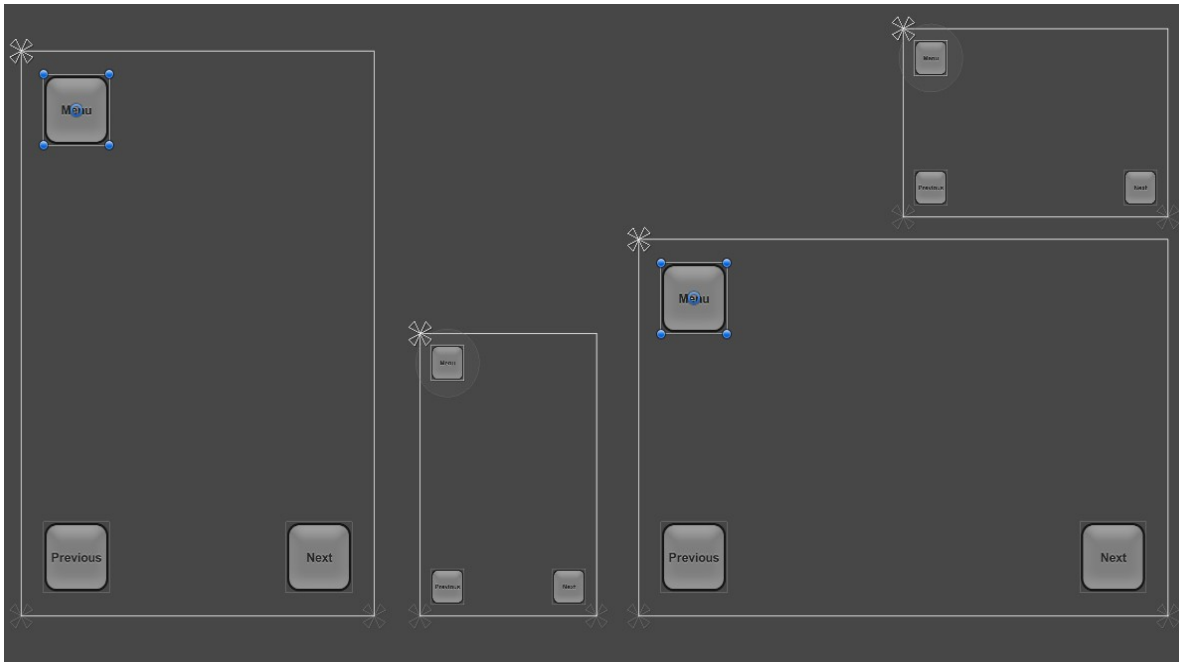
Сучасні ігри та програми часто потребують підтримки широкого спектра різних дозволів екрану, і особливо в даній можливості потребують інтерфейси цих ігор. Система створення інтерфейсів в Unity забезпечена рядом різних інструментів для реалізації цих можливостей, які також можна комбінувати між собою масою різних способів.

Одним із способів зберегти розташування кнопок в області екрана є зміна компонування таким чином, щоб місця їх розташування були пов’язані з їх відповідними кутами на екрані (рис. 18). Прив’язка лівої верхньої кнопки може бути також встановлена в лівому верхньому кутку при використанні в інспектора списку Anchors Preset (набори прив’язок), або шляхом перетягування трикутних ручок прив’язок у видовому вікні сцени (Scene View).

Краще зробити це поки поточний дозвіл екрана, встановлене в ігровому режимі (Game View) є тим дозволом, для якого спочатку все і було задумано, де місця розташування кнопок були б підібрані більш розумно і як говоритися до місця. (Ознайомтеся зі сторінкою UI Basic Layout для отримання більш докладної інформації по прив'язки.). Так само наприклад прив'язки для лівої нижньої і правої нижньої кнопок можуть бути виставлені в лівий нижній і правий нижній кут відповідно.

Як тільки кнопки були прив'язані до своїх кутках, то при подальших змінах дозволів екрану і співвідношень сторін вони будуть зберігати свою позицію щодо цих кутів.

Чого варто побоюватися: так це того, що після додавання компонента Reference Resolution, важливо також перевіряти як буде виглядати компоновка з іншими співвідношеннями сторін. Встановивши дозвіл назад в Phone HD альбомне, можна помітити як кнопки стали більше, ніж повинні бути (і для чого повинні були бути використані) [18][1)].



[1)] [18] Оптимизация Unity UI / Блог компании FunCorp / Хабр. URL: https://habr.com/ ru/company/funcorp/blog/470608/ (дата звернення 12.11.2019).

Рисунок 18 – Інтерфейс оптимізації під різні екрани

2.6 Опис використання сторонніх ресурсів у движку Unity

Моделі і текстури, що зберігаються у вигляді окремих файлів, можна додати в Unity просто перетягнувши їх у вікно Project попередньо створивши в ньому папку Resources, Unity використовує цю папку для завантаження ресурсів за замовчуванням.

Після того як ресурси додані в проект, потрібно вказати в інспекторі для кожного окремого виду ресурсів як і з якими налаштуваннями вони будуть використовуватися.

2.6.1 Опис використання спрайтів у движку Unity

Формат зберігання растрових зображень кадрів анімації або спрайтів буває різний, в залежності від завдання. Поодинокі елементи або кадри анімації, що складаються з декількох спрайтів, зазвичай групують в один файл під назвою атласом. Це дозволяє скоротити кількість використовуваної під них пам’яті, скоротити обсяг займаний проектом, поліпшити продуктивність графічної підсистеми [19][1)].

У нашому проекті спрайт будуть використовуватися як елементи інтерфейсу користувача (рис. 19), кнопки і піктограми меню. Все спрайт зібрані в одному файлі в форматі PNG. PNG є растровий формат зберігання графічної інформації, що підтримує можливість стиснення даних без втрат.

Після того як ми додали файл зі спрайтами в проект, треба вказати Unity як ми будемо його використовувати. Для цього вибираємо його у вікні проекту і потрапляємо в інспектор.

Налаштування, на які слід звернути увагу (рис. 20):

– Texture Type. Тут для Unity слід вказати що з себе представляє цей

---

1)[1)] [19] Форум для разработчиков игр. URL: http://www.gamedev.ru/forum/ (дата звер-нення 25.11.2019).

файл. В даному випадку це Sprite (2D and UI);

– Sprite Mode. Ця опція повідомляє Unity про те, що цей файл зберігає лише одне зображення спрайту, тобто Single, або, як у нашому варіанті, їх кілька, і ми вибираємо Multiple;
– Filter mode. Дозволяє вибрати алгоритм згладжування або фільтрації, що візуально підвищує якість зображення, але погіршує продуктивність;

Далі натискаємо Sprite Editor (рис. 21) і переходимо в редактор спрайтів.

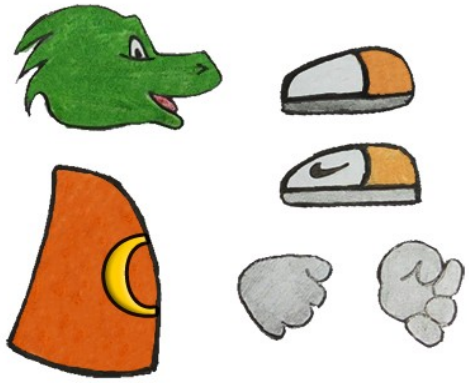


Рисунок 19 – Спрайт

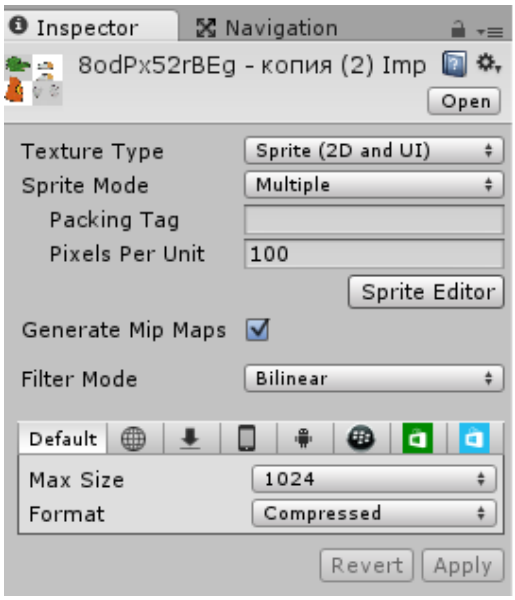


Рисунок 20 – Інтерфейс вікна «Inspector Sprite» у движку Unity

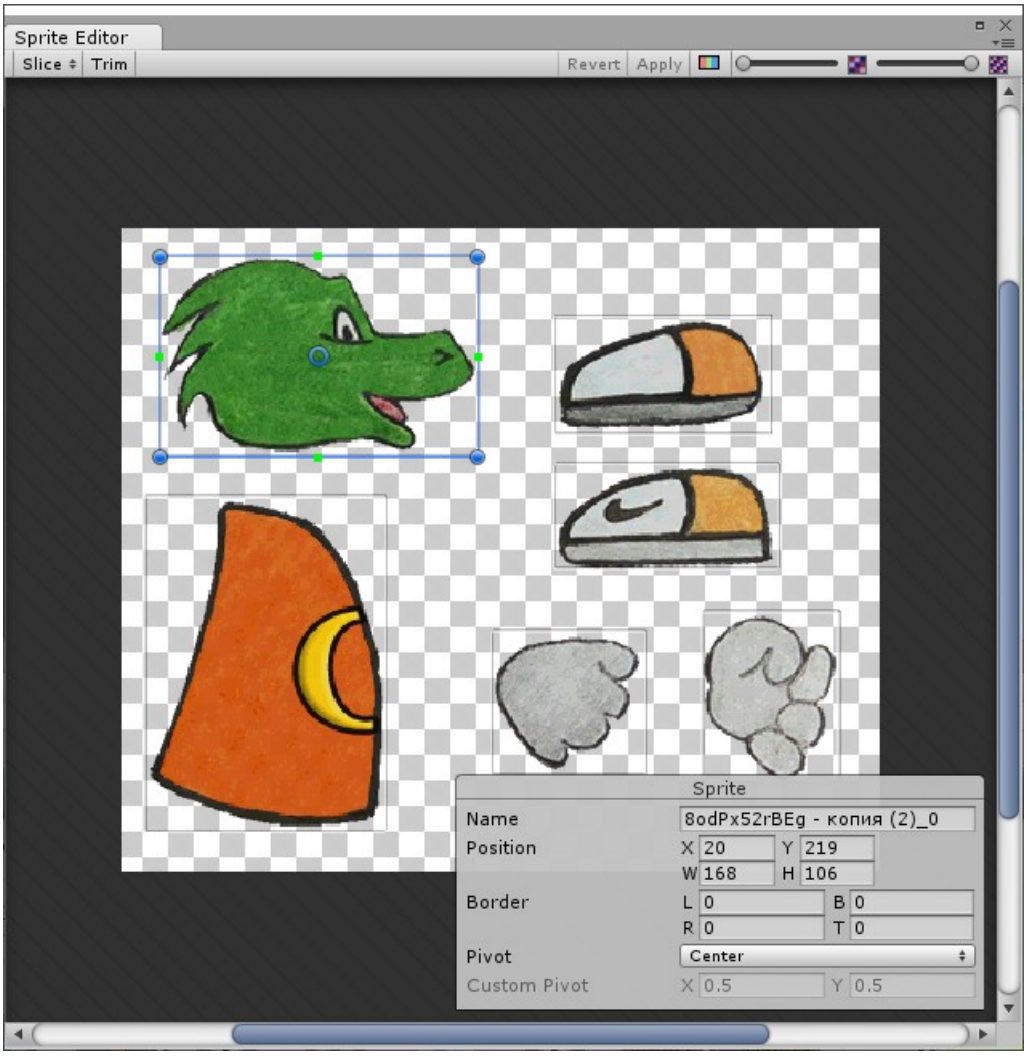


Рисунок 21 – Інтерфейс редактора спрайтів Unity

Тут ми скористаємося автоматичної нарізкою об’єктів Slice. Unity сама виділить кожен елемент на зображенні і дасть згодом можливість нам використовувати його як окремий спрайт, Unity сама буде підставляти координати цього елемента в атласі для відображення його на сцені. У разі якщо межі об’єктів не чітко видно або з якихось причин нам не підходить варіант автоматичної розмітки, ми можемо самі вказати розміри об’єктів, відстані між ними і розмітити самостійно.

Після нарізки, у файлу з спрайтами у вікні проекту з’являться вкладені

дочірні елементи (рис. 22). Тепер у нас є можливість вибрати окремий об’єкт в атласі і використовувати його в дизайні елементів меню користувача.

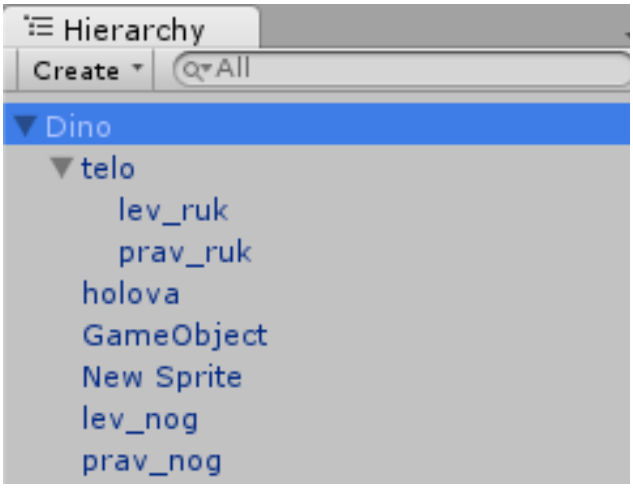


Рисунок 22 – Інтерфейс вікна «Hierarchy» у движку Unity з елементами спрайта

2.6.2 Використання шрифтів у движку Unity

Технологія по додаванню в проект аналогічна спрайт. Ніяких особливих налаштувань в нашому випадку чіпати не потрібно, все можна залишити за замовчуванням. Шрифт нам буде потрібно для створення написів на кнопках в головному меню. Ми будемо використовувати безкоштовний векторний шрифт Carbon Black, створений компанією Typodermic fonts.

3 РОЗРОБКА ТА ОПТИМІЗАЦІЯ ІНТЕРФЕЙСУ ДЛЯ ІГРОВОГО ДОДАТКА НА ПЛАТФОРМІ ANDROID

3.1 Розробка моделей для ігрового додатка

Розробка моделі зайняла 3 етапи. Першим етапом було візуалізація на папері. Другий етап був сканування і обробка в програмі Adobe Photoshop. (рис. 23). Третім етапом була робота в Unity

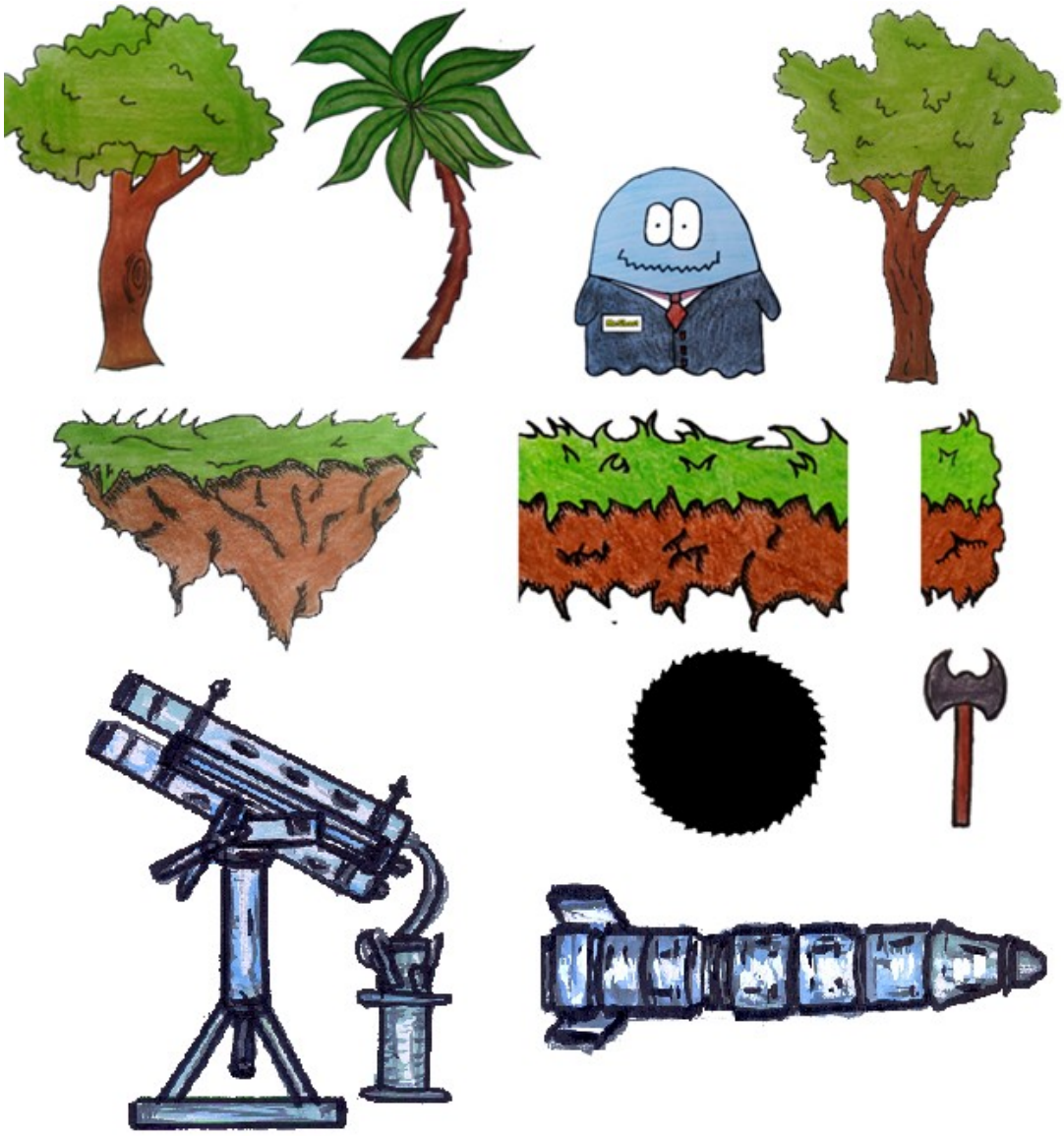


Рисунок 23 – Моделі після обробки

Найскладнішим у проекті був головний персонаж. Головний персонаж складається з 6 окремих частин, таких як голова, тіло, 2 руки та 2 ноги. Ці окремі частини (спрайти) у проекті з’єднуються та анімуються (рис. 24).

"Спрайт" - 2D графічні об'єкти, що використовуються для персонажів, бутафорії, снарядів та інших елементів 2D геймплея. [20][1)].

Рисунок 24 – Модель головного персонажа

1)1) [20] Unity - Руководство: Sprite Creator URL: https://docs.unity3d.com/ru/current/-Manual/SpriteCreator.html (дата звернення 25.11.2019

3.2 Робота з анімацією у движку Unity

Анімація – вид мультиплікації, створюваний за допомогою комп’ютера. На відміну від більш загального поняття «графіка CGI», що відноситься як до нерухомих, так і до рухомих зображень, комп’ютерна анімація має на увазі тільки рухомі [21][1)].

У даному проекті анімація є на таких об’єктах, як головний персонаж, пружина, сніговик та привід.

Анімація працює за принципом захоплення кадру на шкалі часу. Привид з’являється і зникає, для цього перший кадр на цій шкалі часу–коли привид є, другий кадр де його немає (рис. 25).

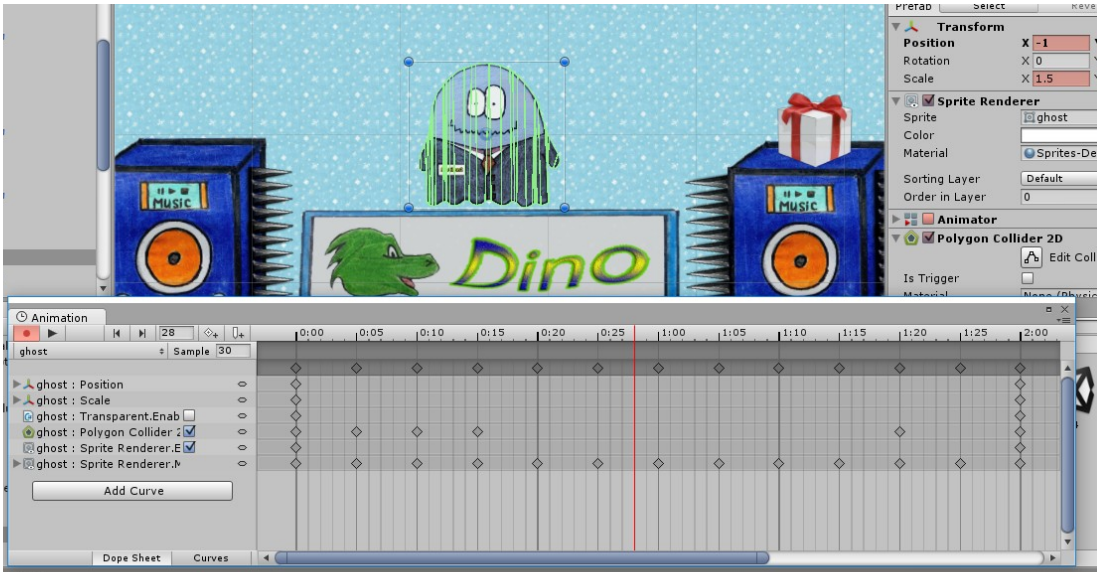


Рисунок 25 – Скриншот анімації «зникнення привида»

Сніговик – другий наш ворог, у нього дві анімації, він погрожує пістолетом а потом робить 3 постріла зі звуковим сингналом (рис. 26) та атакує мітлею (рис. 27). Було також у режимі «Аніматор» створено перехід між анімаціями (рис. 28).

---

1)[1)] [21] 2d анимация в unity - Your Game Now. URL : https://3dgame-creator.ru/ unity-5-2d-animaciya/. (дата звернення 30.11.2019).

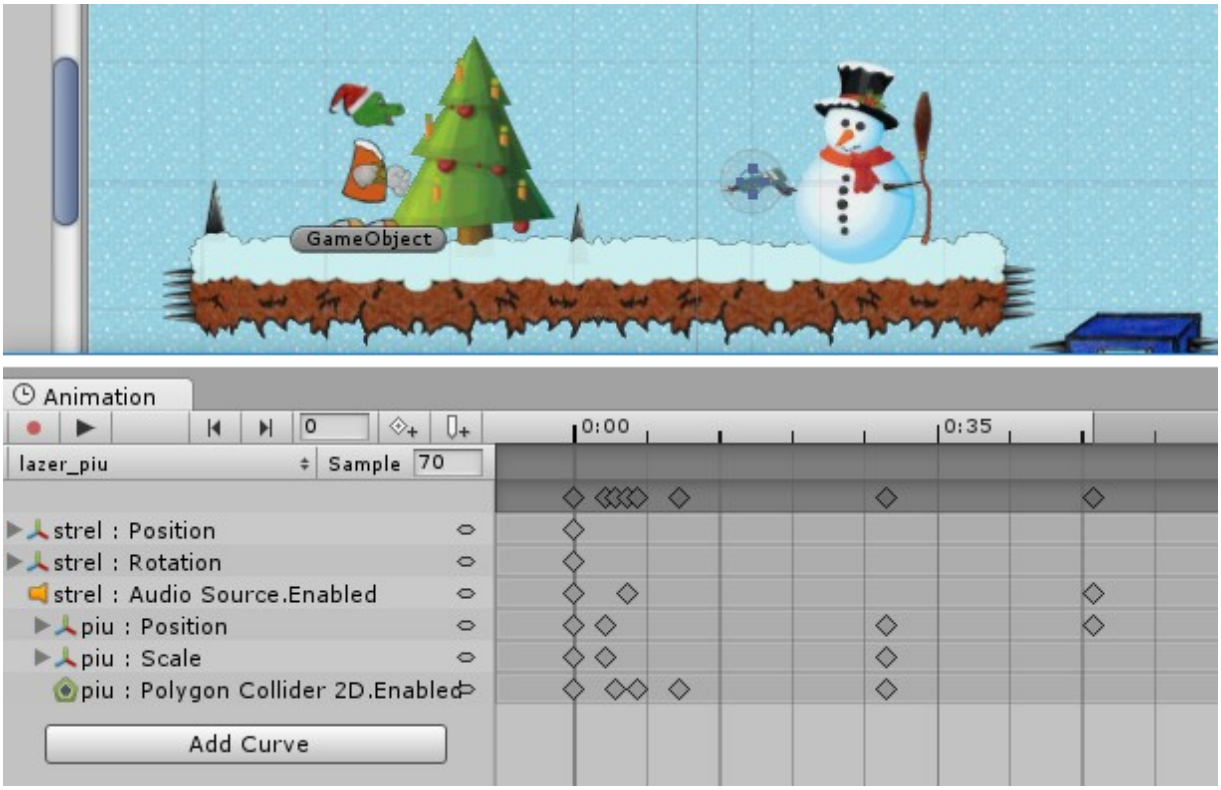


Рисунок 26 – Скриншот анімації «пострілу сніговика»

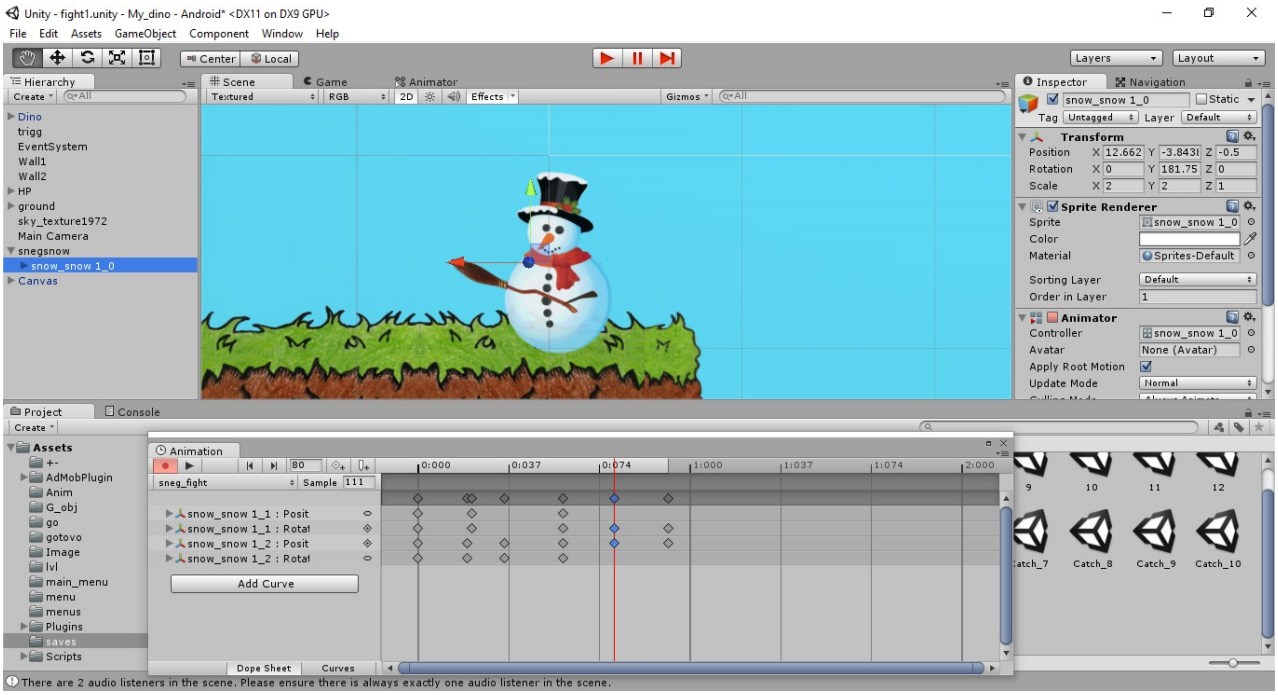


Рисунок 27 – Скриншот анімації «удара сніговика»

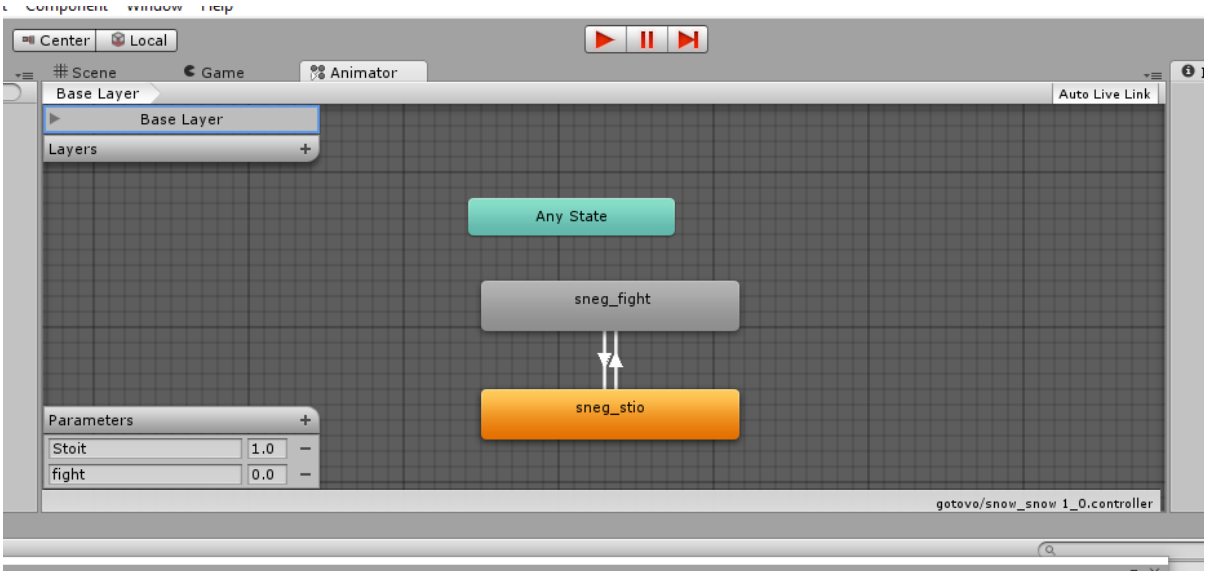


Рисунок 28 – Скриншот умови для анімації

Пружина служить щоб персонаж мав можливість підпригувати на інші блоки. Тут зроблена анімація стиснення пружини та звуковий сигнал (рис 29).

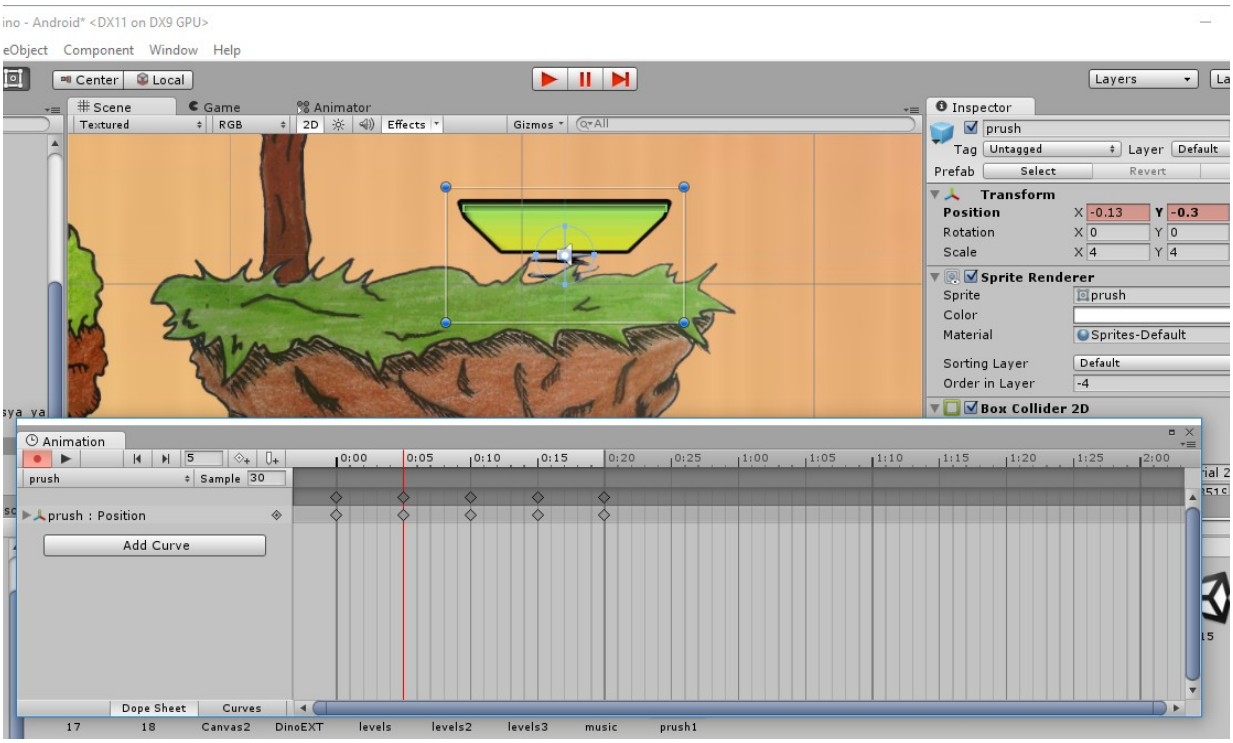


Рисунок 29 – Скриншот анімації пружини

Наступним чино була зроблена анімація для ракети. При різному погодному стані, гравець повинен вибрати правильну ракету. Усього їх 3. Перша запускається коли недалеко від населеного пункта, тварин, людей полів, садів. Наступна запускається коли далеко від населеного пункта, села, садів, полів. Третя запускається коли треба зовсім недалеко запустити для усунення погодного стану.

Тому було зроблено 4 анімації, а саме:

– перша ракета (рис 30);
– друга ракета (рис 31);
– третя ракета (рис 32);
– стан за замовчуванням.

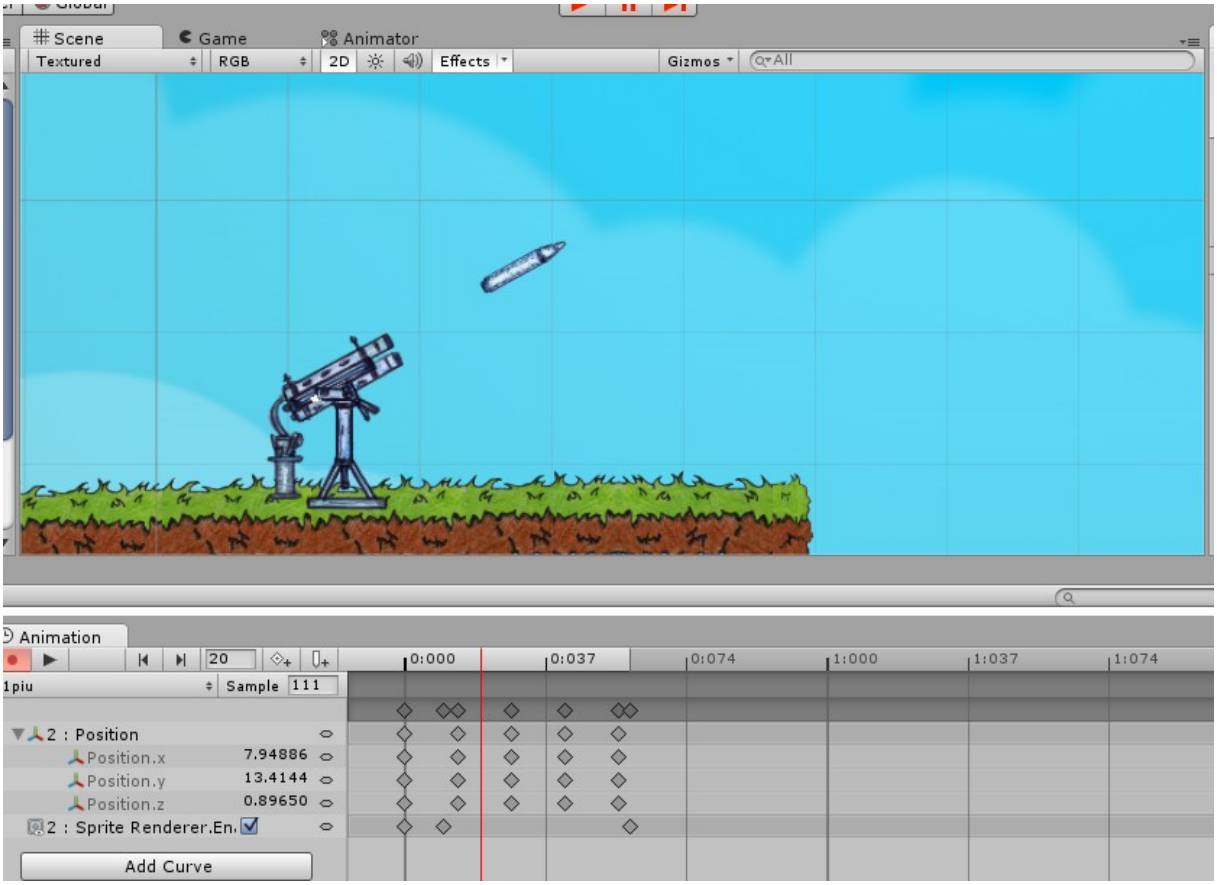


Рисунок 30 – Скриншот анімації «запуск першої ракети»

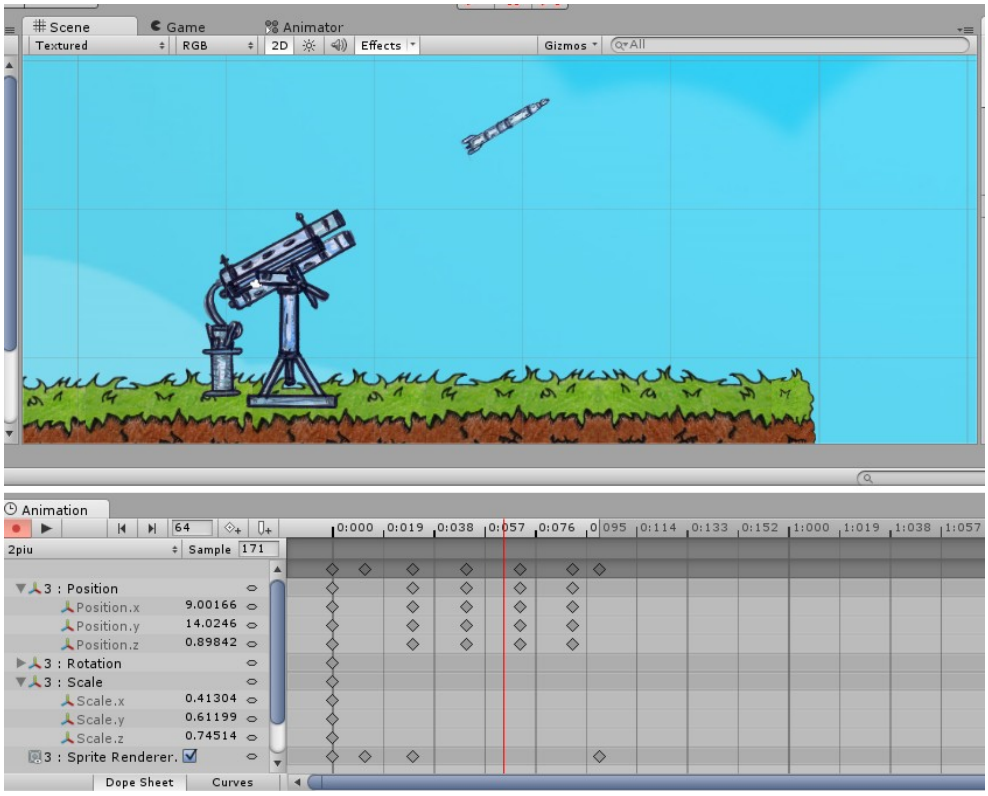


Рисунок 31 – Скриншот анімації «запуск другої ракети»

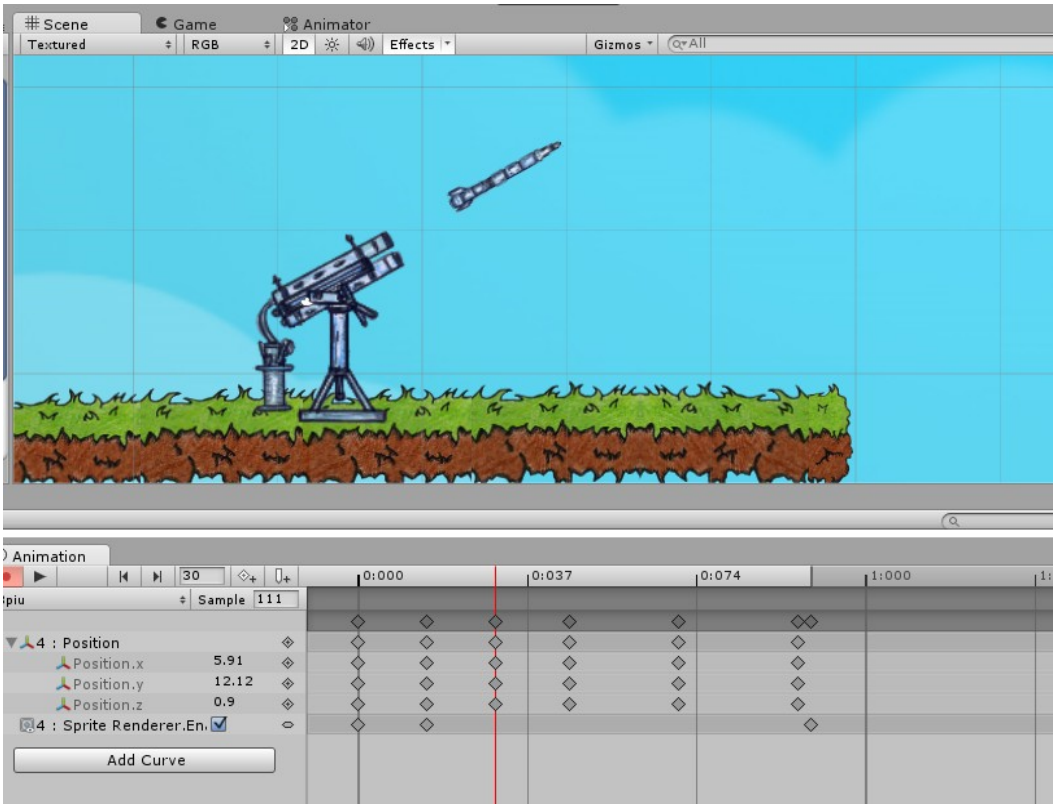


Рисунок 32 – Скриншот анімації «запуск третьої ракети»

За допомогою «Animator» було зроблено умови для переходу між анімаціями. Коли перша змінна дорівнює «1», то зі стана за замовчуванням анімація буде переходити до анімації запуска першої ракети. Після цього анімація автоматично переходить до стана за замовчуванням. Таким же принципом було зроблено інші ракети (рис. 33).

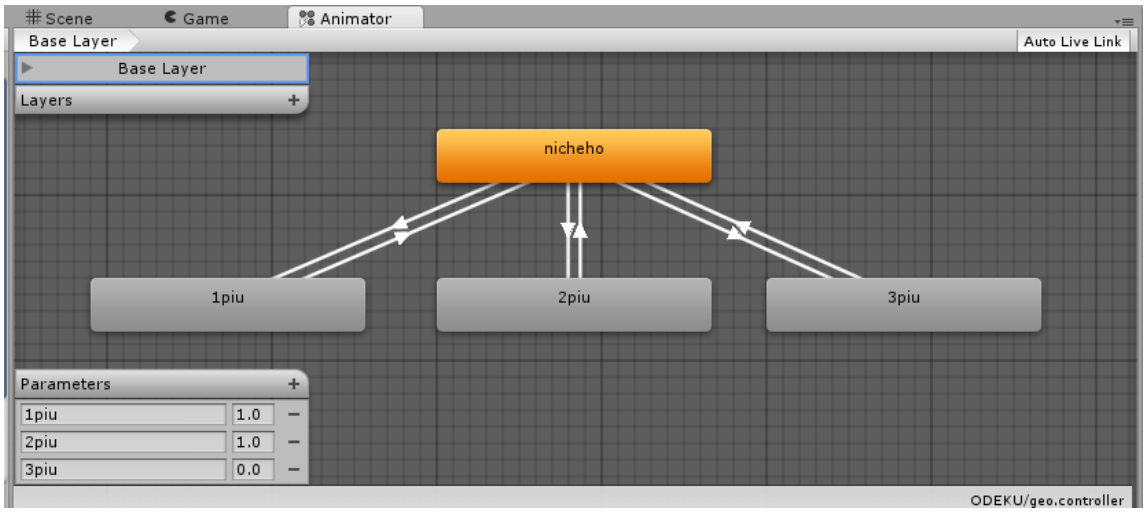


Рисунок 33 – Скриншот вікна «Animator»

Для того щоб анімація відображалася максимально правильно, було налаштовано перехід між анімаціями вручну (рис. 34).

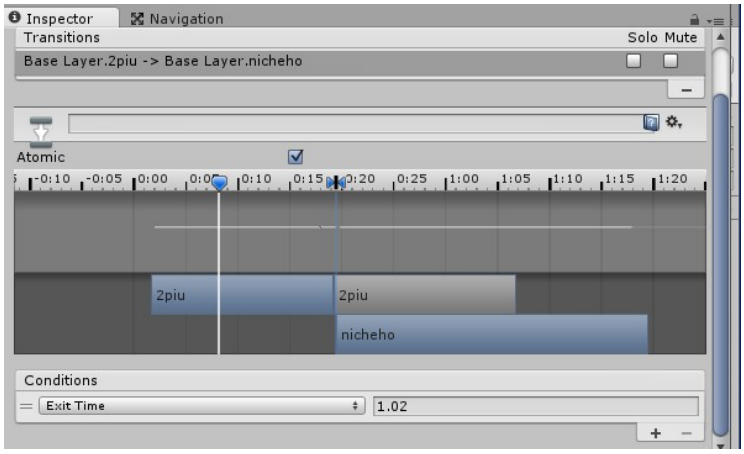


Рисунок 34 – Скриншот налаштування анімації

Сама важка анімація – анімація головного героя , адже у нього їх аж 4. Це коли він стоїть (рис 35), коли йде (рис 36), коли бореться (рис 37) та коли присідає (рис 38), для цього були зроблені умови, при яких анімація буду змінюватися(рис 39)

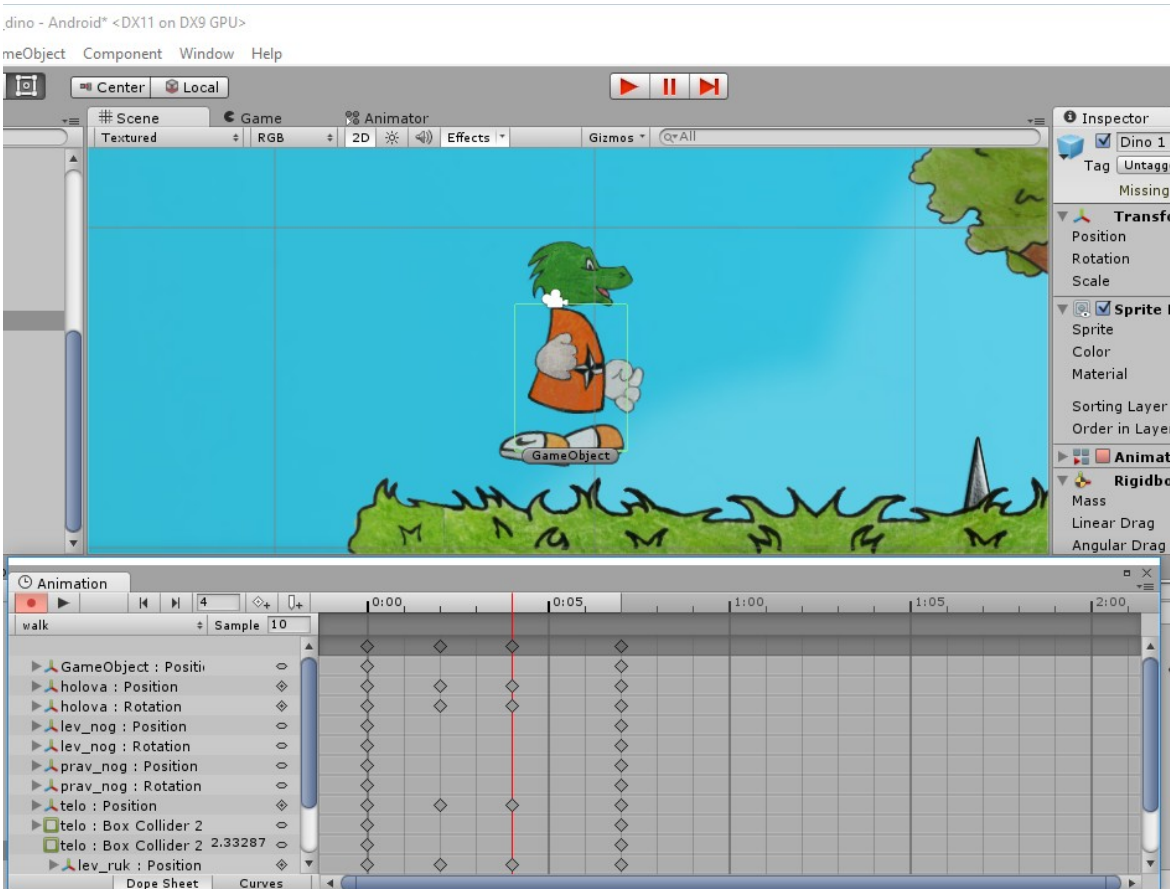


Рисунок 35 – Скриншот анімації «Персонаж стоїть»

Зберіг всі файли анімації, та скористався засобом "аніматор" в Unity для того, щоб задати можливість головному персонажу міняти анімацію в залежності від того, як буде розпоряджуватися ним користувач. Створив новий слой і перетягнув туди всі анімації головного персонажа.

Було зроблено різні параметри, як «йде», «бореться», «присідає» та «стоїть». За замовчуванням персонаж буде виконувати анімацію "стояти на місці", коли значення "1" у полі параметри, буде замінено, то буде відбуватися інша анімація, поки цифра не стане знову одиницею.

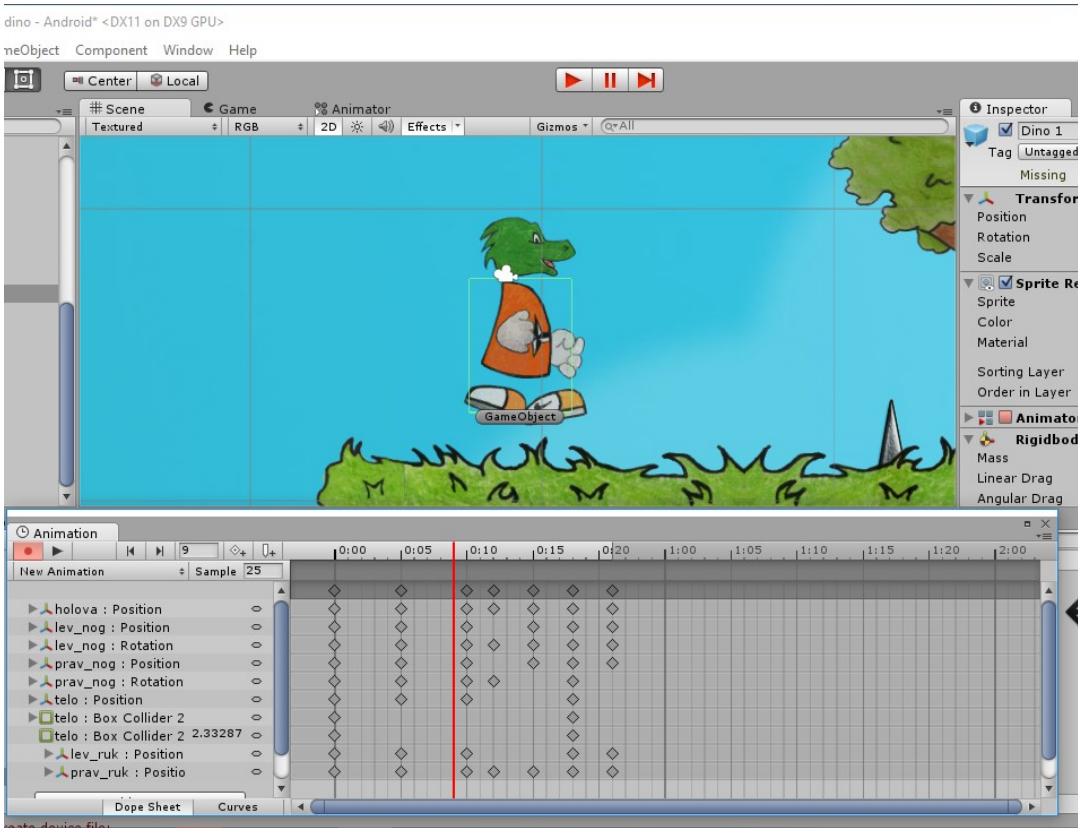


Рисунок 36 – Скриншот анімації «Персонаж йде»

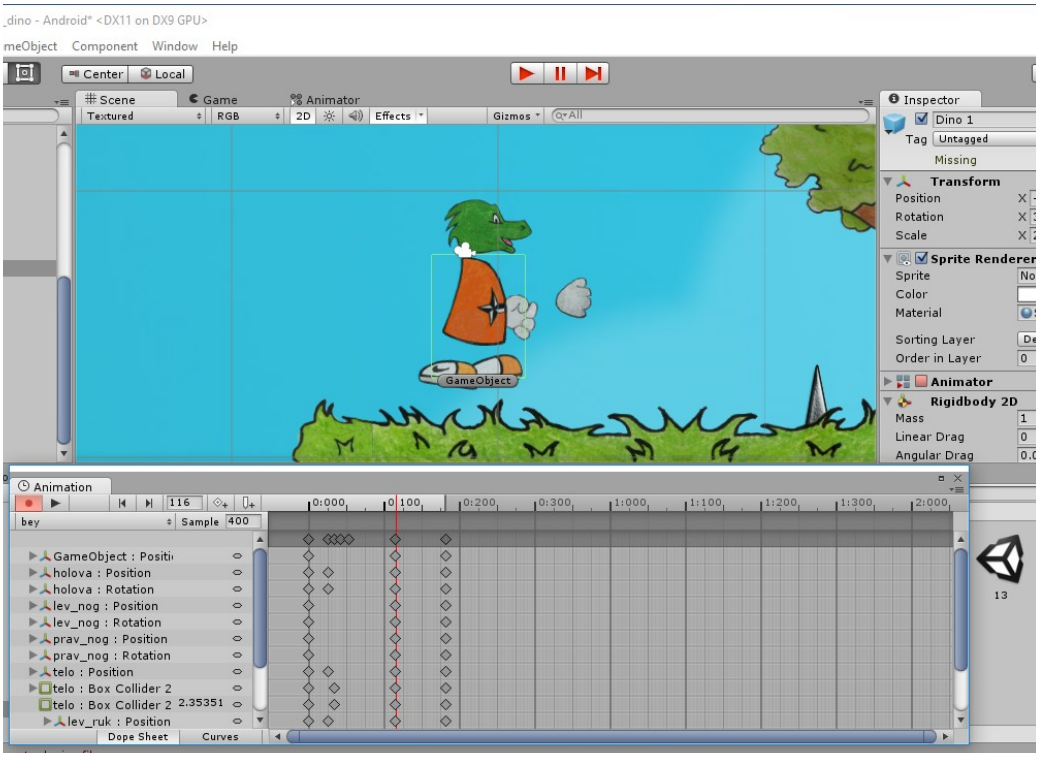


Рисунок 37 – Скриншот анімації «Персонаж бореться»

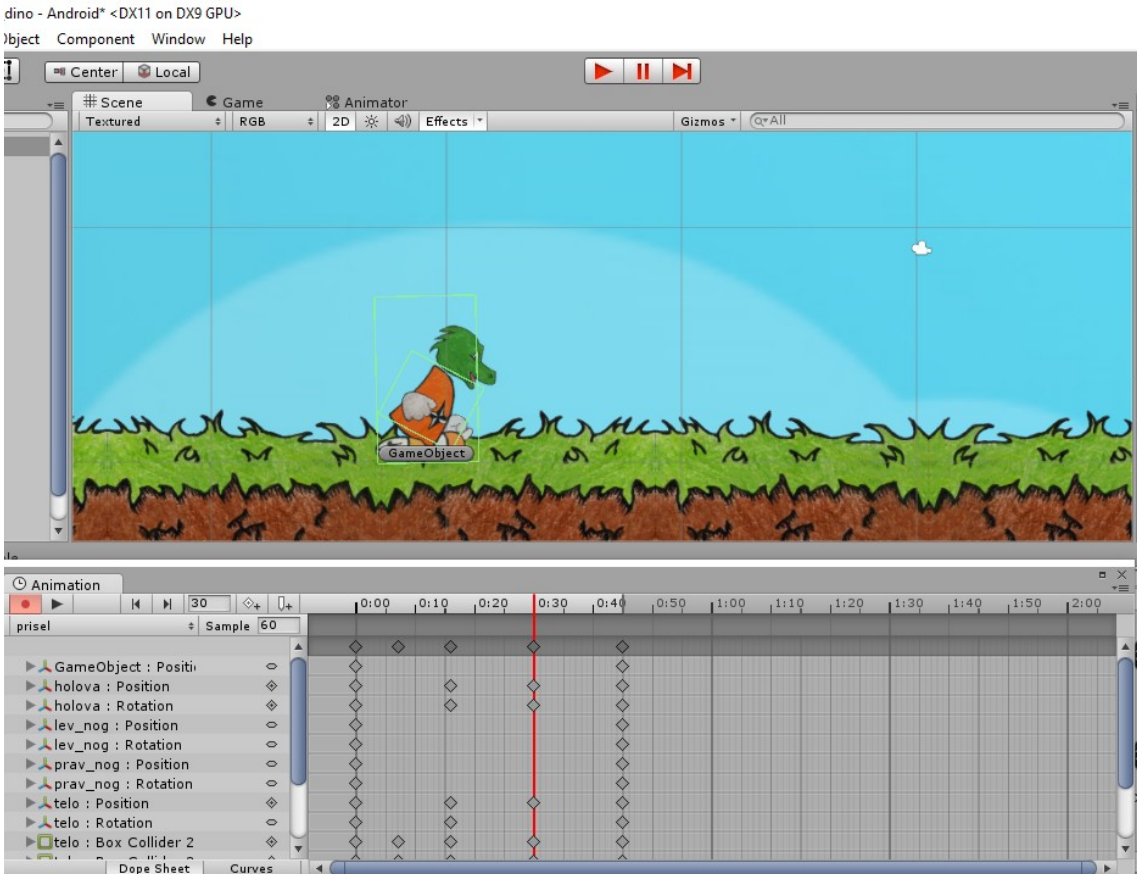


Рисунок 38 – Скриншот анімації «Персонаж присідає»

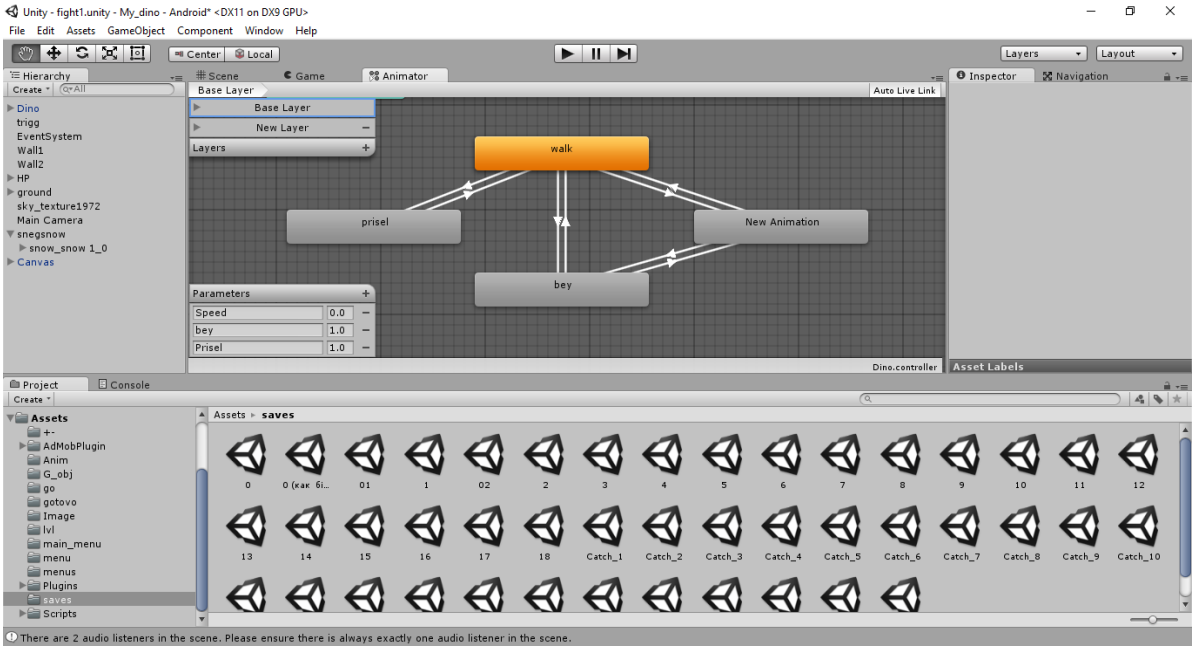


Рисунок 39 – Скриншот умов для анімації

Коли відбувається заміна анімації, вона не повинна різко мінятися, не закінчивши стару, для цього було зроблено плавні переходи для зміни кожної анімації (рис. 40).

Для того, щоб анімація вийшла максимально акуратною, запускалася допоміжна форма , у якій була можливість вибрати модель, яку ми бажаємо анімувати і в режимі реально часу змінювалися час анімації, а за ними і план переходу анімації з однієї в іншу.

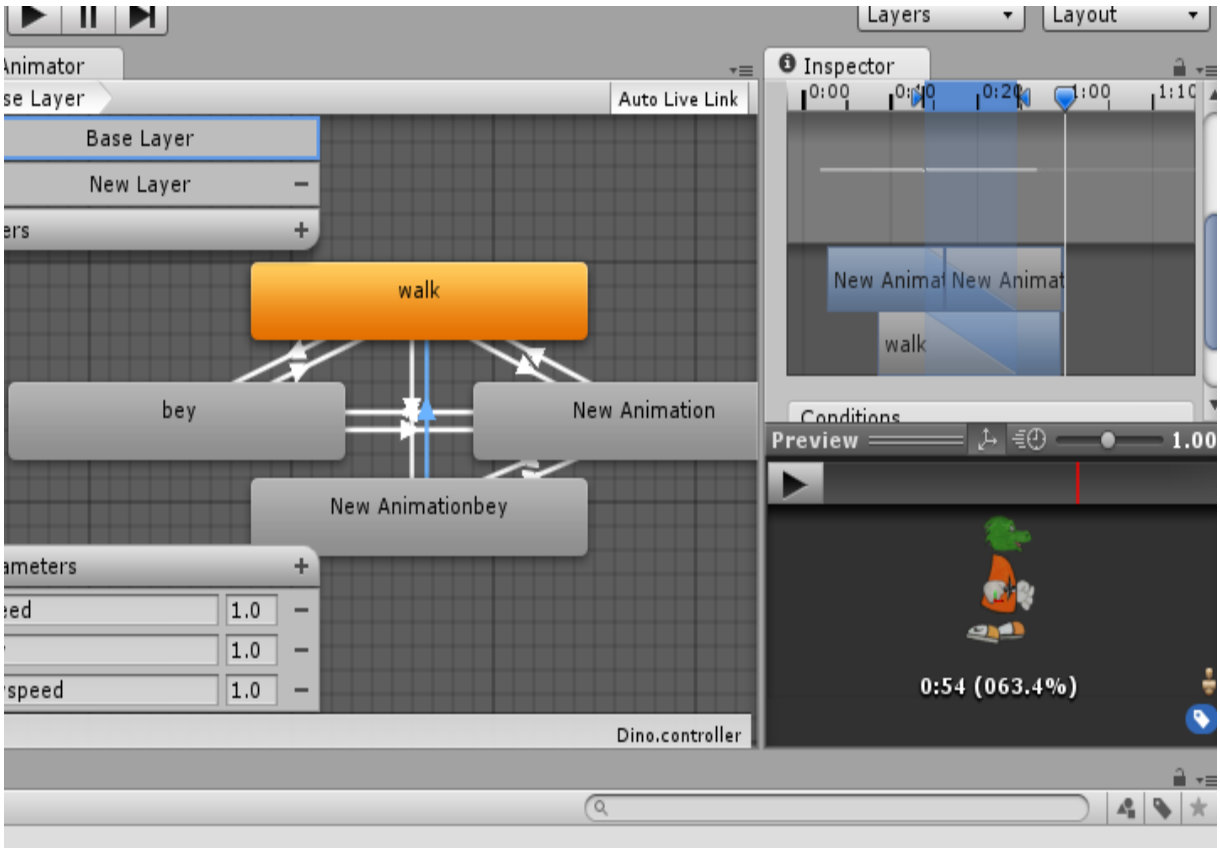


Рисунок 40 – Скриншот переходів при зміні анімації

Зробивши плавний перехід в будь-який з можливих випадків, була вибрана основа анімація, котра буде основною та було зафіксовано анімацію на персонажа в його властивості управління та після цих дій, головний персонаж може бути експортован на сцену для подальшої розробки (рис 41).

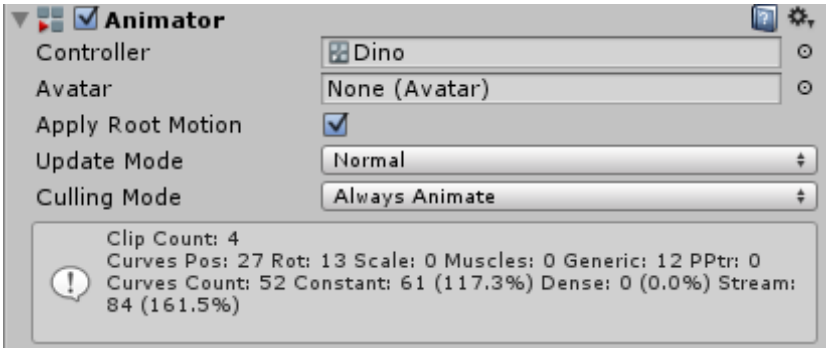


Рисунок 41 – Скриншот фіксації анімації головному персонажу

3.3 Робота з фізикою, оптимізація та розробка сцен у движку Unity

Для того, щоб персонаж мав можливість передвигатися, стрибати, присідати і не падати за межі карти, йому було надано масу, та виділено коллайдером ділянки тіла. Rigidbody – це основний компонент, що включає фізичне поводження для об’єкта [22][1)]. З прикріпленим Rigidbody, об’єкт негайно почне реагувати на гравітацію. Якщо доданий один або кілька компонентів Collider, то при колізіях (зіткненнях) об’єкт буде пересуватися. Форма коллайдера задається шляхом вільного визначення граней сегментів ліній, з яких вона складається. Колайдери це інший тип компонентів, які повинні бути додані разом з твердими тілами, щоб задіяти зіткнення [23][2)]. Якщо два твердих тіла врізаються один в одного, фізичний движок не буде прораховувати зіткнення, поки до обох об’єктів не буде призначений коллайдер, а тобто коллайдер допомагає персонажу взаємодіяти з іншими предметами (рис 42).

Базовим елементом користувальницького інтерфейсу Unity є Canvas (полотно). Він відповідає за генерацію, сортування і отрисовку мешів дочірніх елементів інтерфейсу. Всі елементи UI повинні бути дочірніми до

[1)1)] [22] Unity - Руководство: Rigidbody 2D (Двухмерное твердое тело). URL : https://docs.unity3d.com/ru/current/Manual/class-Rigidbody2D.html. (дата звернення 02.12.2019).

[2)2)] [23] Unity - Руководство: Коллайдеры (Colliders) https://docs.unity3d.com/ru/current/Manual/CollidersOverview.html (дата звернення 04.02.2019).

якогось полотну, в іншому випадку вони не будуть відображатися в грі.

Для даної гри було вирішено створення 4 міні-гри, таких, як "пригода", "ранер", "ухилення", та "ОДЕКУ", тому було зроблено головне меню, вибір гри та вибір рівня за допомогою інтерфейса «Canvas». Він служить щоб усі об’єкти було розташовані по розміру екрана на смартфоні (рис 43).

За допомогою інтерфейса «Canvas» було також зроблено управління для гри, такі як, присісти, рух вправо, вліво, стрибок та атака (рис 44). Інтерфейс «Canvas» служить помічником для того, щоб на кожному девайсі на платформі андроїд було розташовано управління максимально зручним, воно не буде занадто великим, чи маленьким і завжди буде розташовано там, де було указано. Цю заготовку я завантажив для того, щоб перекинути на інші рівні. За допомогою цього я зміг оптимізувати кнопка управління під кожен екран.

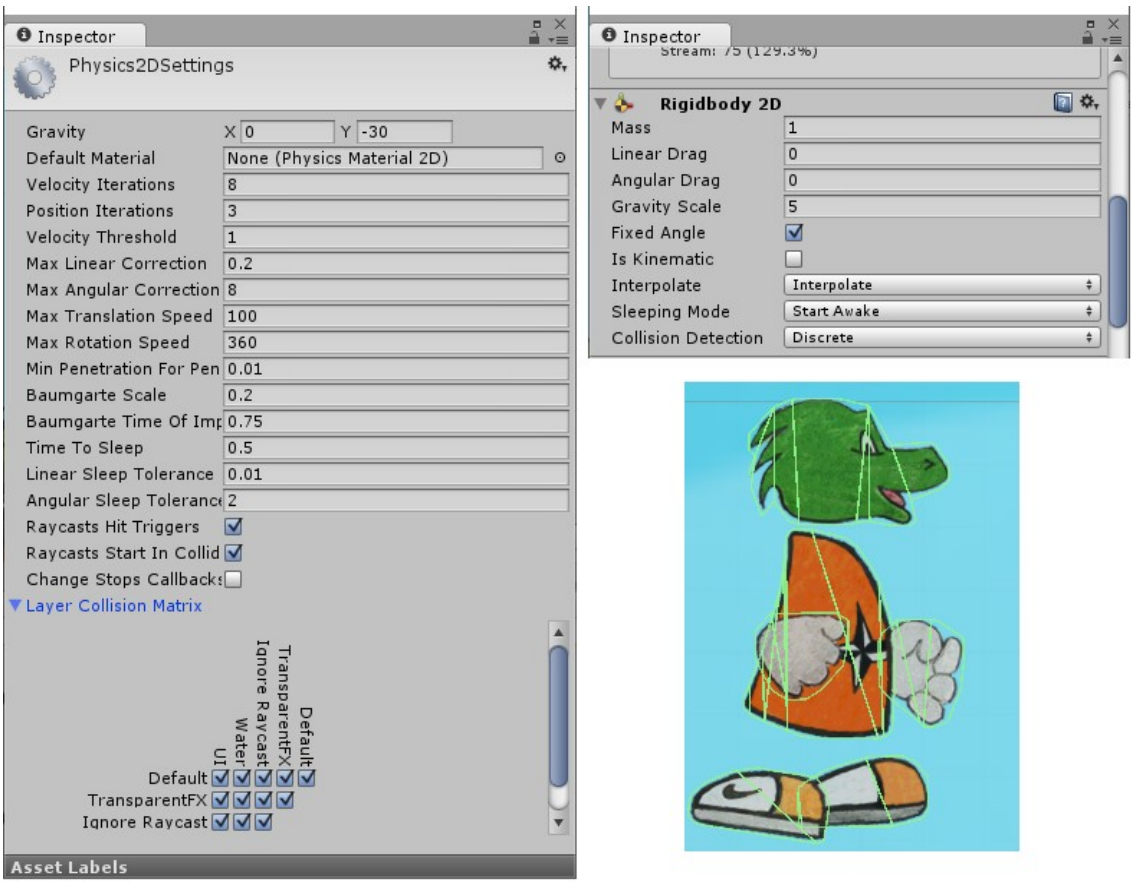


Рисунок 42 – Скриншот налаштування фізики для гри

Рисунок 43 – Інтерфейс головного меню та вибір рівня

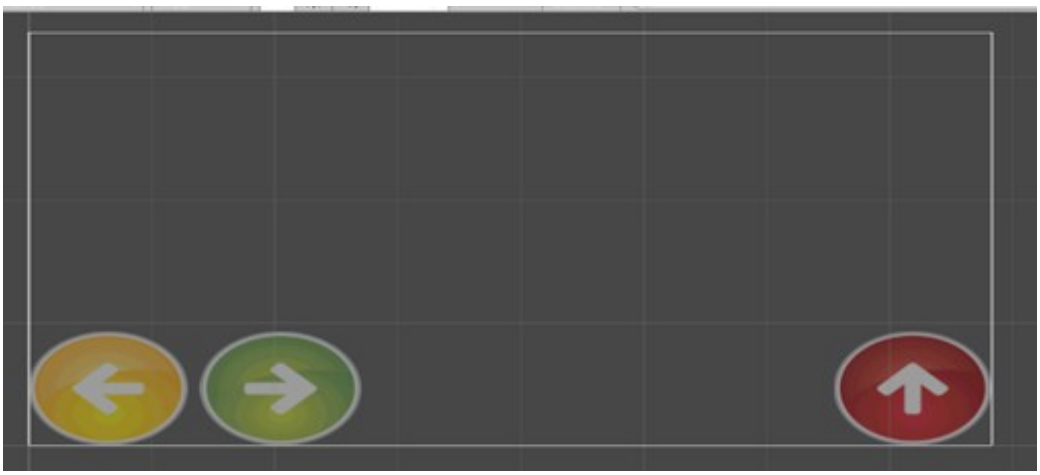


Рисунок 44 – Інтерфейс кнопок управління

Було задано коллайдер на кожен з фізичних об’єктів, таких, як платформа землі, пила, шипи, антагоністи та збережено для експорту на карту.

Наступнім етапом була змодельована перша локація для гри «adventures». Для того, щоб кожна модель була на своєму місці, їй потрібно задати позицію не тільки по осям x та y, а і указати позицію z для того, щоб кожен спрайт був на указаному місці. Суть гри полягає у зборі усіх бонусів, які попадаються на путі, але треба не потрапить у пастку. Кожен наступний рівень було змодельовано так, щоб гравцю було тяжче та цікавіше його проходити (рис. 45). Усього було створено 18 сцен для цього режима.

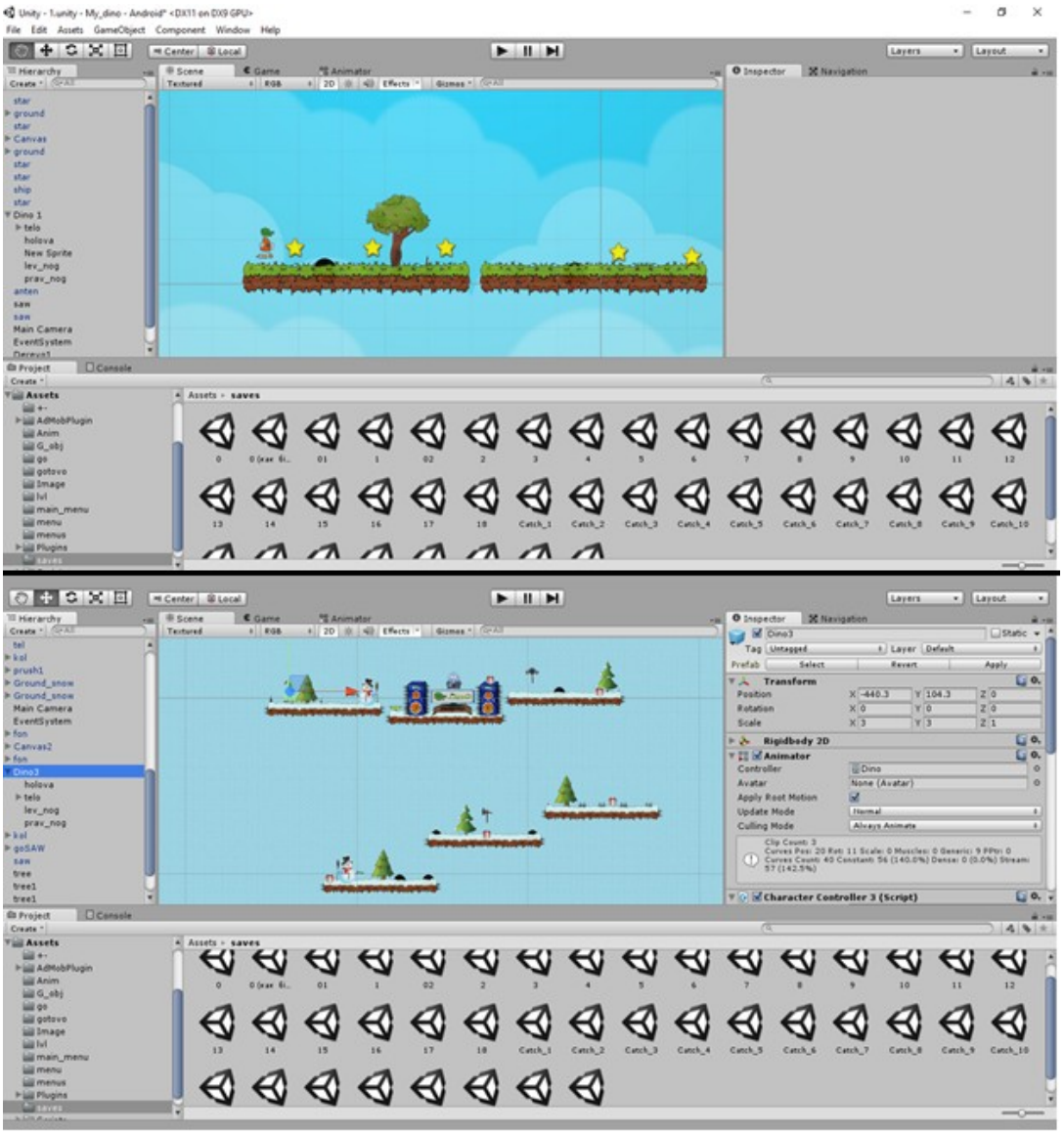


Рисунок 45 – Скриншот локацій у режимі «adventures»

Після того, як було змодельовано перший режим, було змодельовано другу локацію для гри «ухилення». Суть гри полягає у тому, що зверху генеруються пастки, бонуси та додаткове життя, які направляються униз до головного героя. Головному персонажу потрібно збирати бонуси та життя та не попадатися в пастки (рис. 46). Таких сцен було створено 12 для цього режиму.

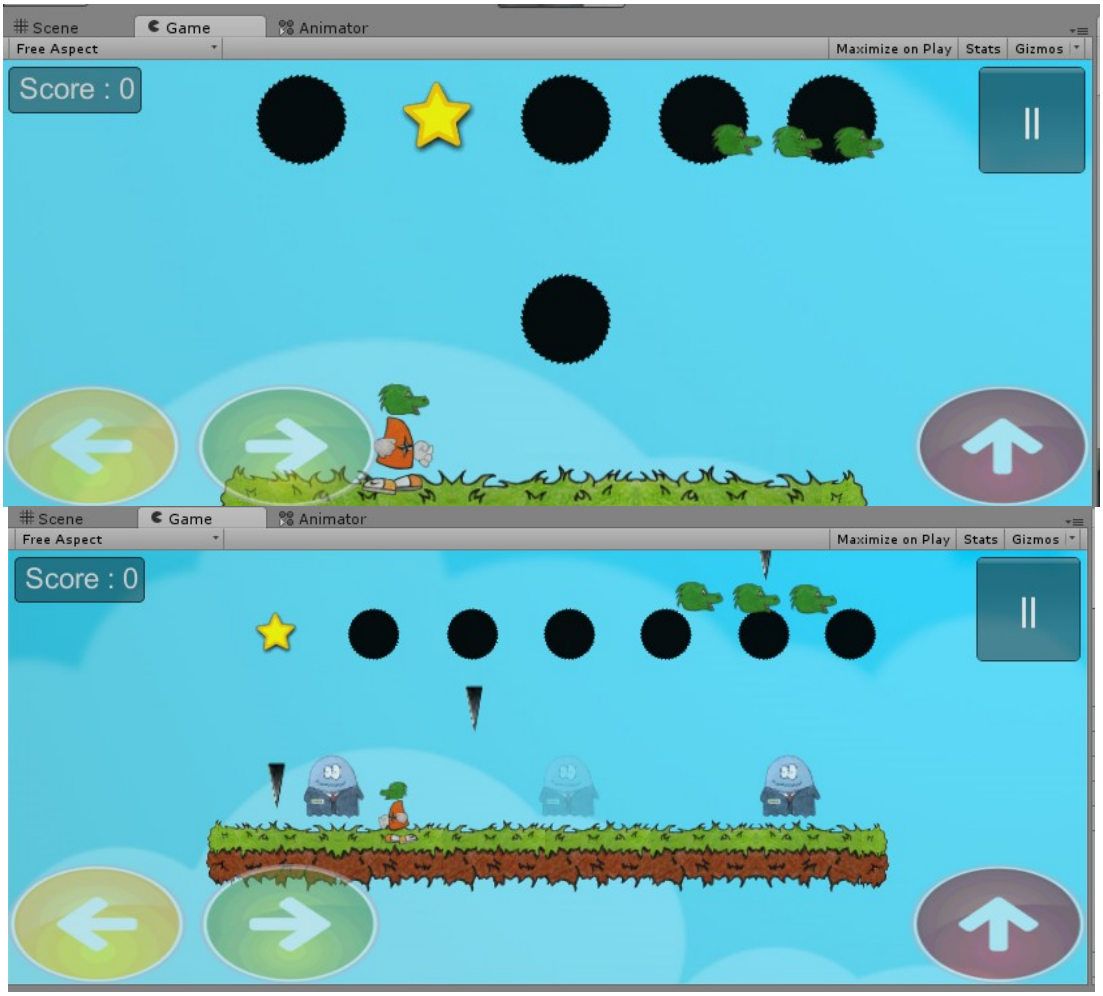


Рисунок 46 – Скриншот локацій у режимі «ухилення»

Наступним етапом було створення локацій для гри «раннер». Суть гри дуже проста, головний персонаж завжди бігає, карта генерується хаотично. Головному персонажу 4 життя, та йому потрібно не попадатися на

пастки та збирати бонуси (рис. 47). Таких сцен було створено 12.

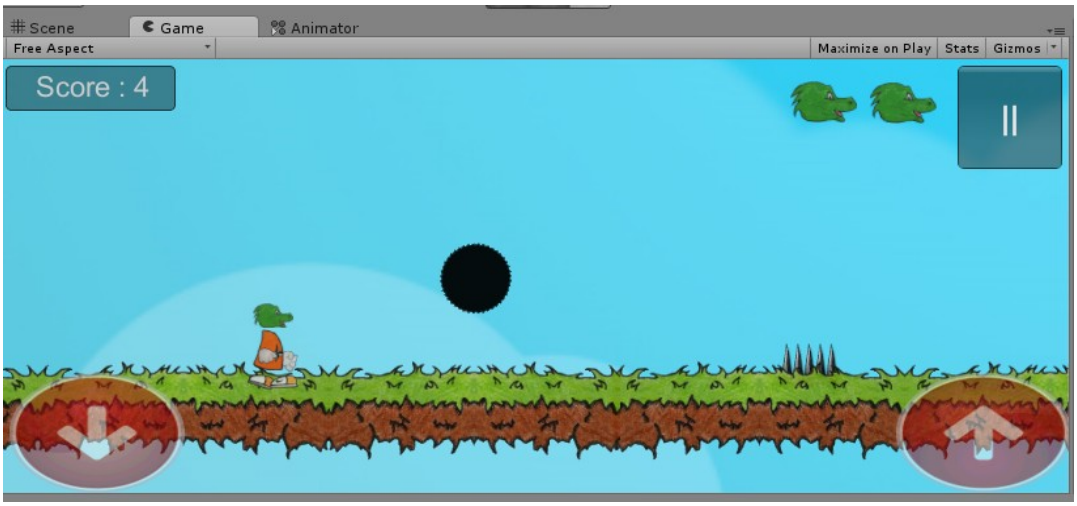


Рисунок 47 – Скриншот локації у режимі «раннер»

Останнім етапом була розроблена фінальна гра, котру було вирішено назвати «ОДЕКУ». Головний персонаж повинен врятувати місто від хмар. Йому в цьому будуть допомагати індикатори, телескоп та вибір правильної ракети. Коли моргає індикатор хмаринки, персонажу потрібно підійти до телескопу (рис. 48) та вияснити які наближаються хмари, як далеко вони від населеного пункта, садів, полів і взагалі як далеко від нас самих, тому що якщо запустити не ту ракету, вона може травмувати когось або зруйнувати щось.

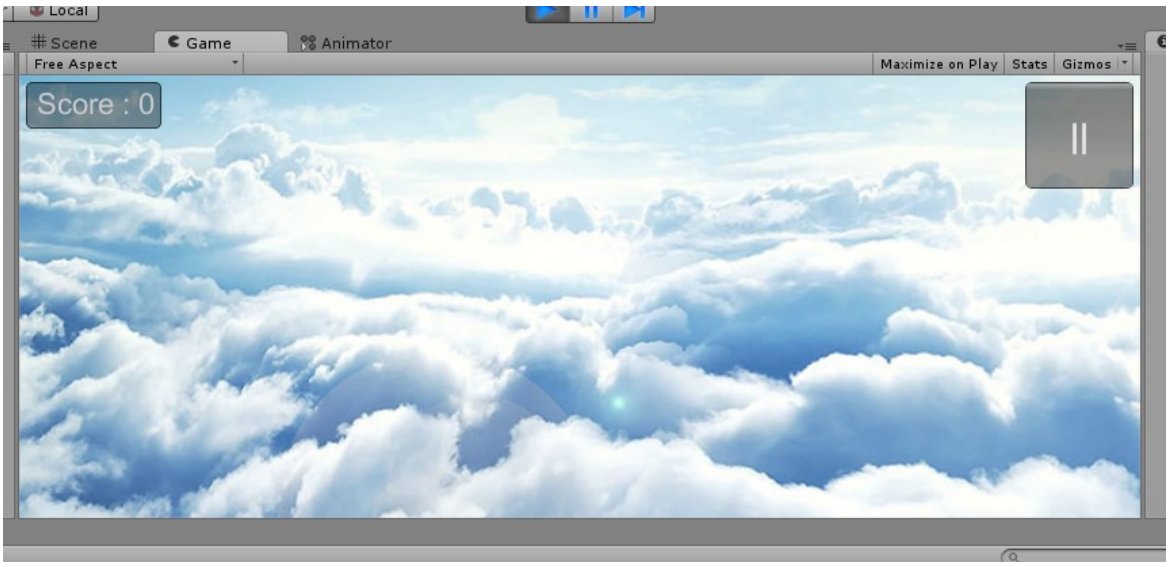


Рисунок 48 – Скриншот виду з телескопа у режимі «ОДЕКУ»

Коли з’ясували які наближаються хмари, персонаж повинен вибрати правильну ракету та запустити її (рис. 49). За кожний заправильний вибір, персонаж получить 20 балів, а за невірний вибір він позбудеться 50 балів.

Рисунок 49 – Скриншот запуску ракети у режимі «ОДЕКУ»

3.4 Експорт ігрового додатка на платформу Android

Першим ділом було завантажено утиліту Java JDK та Android SDK. Після установки було завантажено, у середовищі Android SDK, SDK платформи для сумісністю з різними девайсами на платформі Android [24][1)] (рис. 50).

Наступним ділом у платформі Unity зайшов в меню файл та вибрав «Build setting» та виставив усі сцени у полі «scenes in build». Тепер, при компіляції ці сцени, що находяться у цьому полі будуть використані. Якщо якусь сцену забувся, додаток не зможе працювати правильно. Усього 50 сцен у

---

1)[1)] [24] Unity - Руководство: Настройка Android SDK: https://docs.unity3d.com/ru/current/Manual/android-sdksetup.html. (дата звернення 04.02.2019).

цьому проекті. У полі «platform» вибрав платформу Android та натиснув на switch platform для того, щоб Unity запам’ятовало мій вибір (рис. 51).

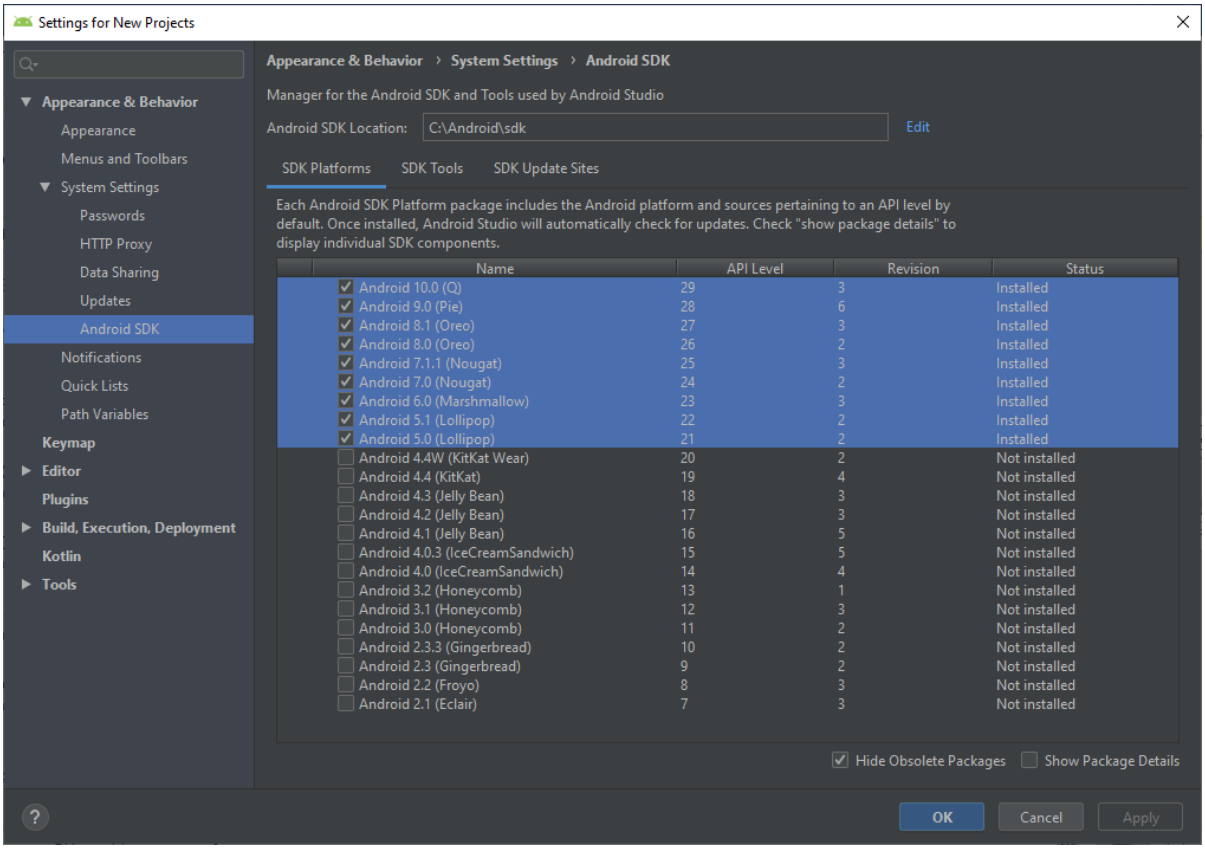


Рисунок 50 – Скриншот завантаження Android SDK

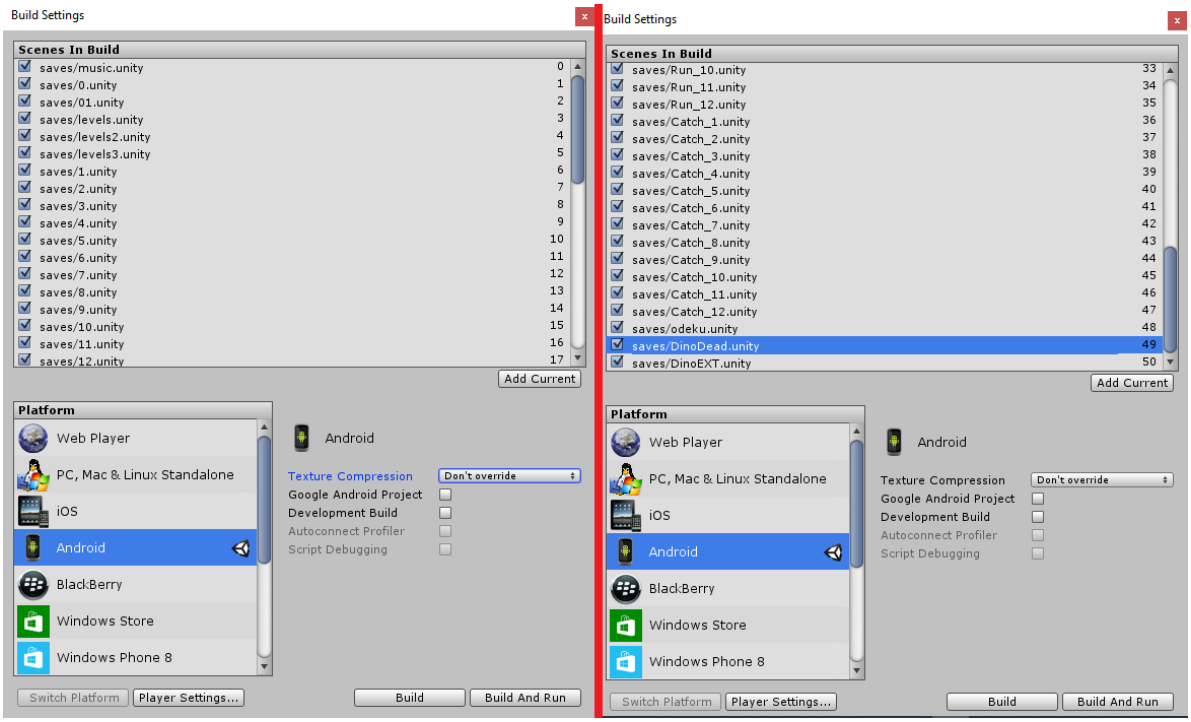


Рисунок 51 – Скриншот налаштування сцен та вибір платформи

Останнім кроком було вибрати пункт «player settings…» та завантажити усі особливі моменти (рис. 52), а саме:

– назва компанії;
– назва продукта;
– міні іконку;
– вибір рендерінга;
– ідентифікація;
– оптимізація;
– приватний ключ.

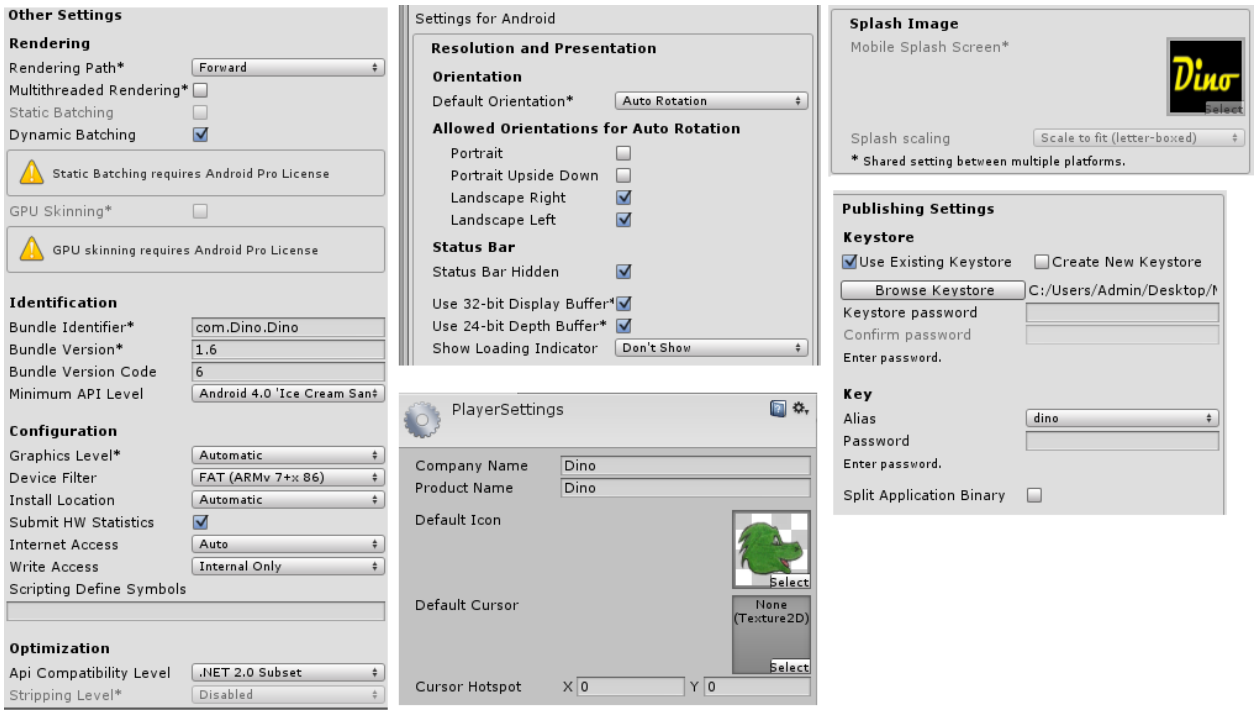


Рисунок 52 – Скриншот налаштування інформації для експорту додатка

Після усіх налаштувань, гру було експортовано у файл з розширенням APK на комп’ютер та завантажена на мобільний пристрій.

4 УПРАВЛІННЯ ПРОЕКТОМ

Проект – унікальний набір процесів, що складаються зі скоординованих і керованих завдань з початковою і кінцевою датами, вжитих для досягнення мети. Досягнення мети проекту вимагає отримання результатів, що відповідають певним заздалегідь поставленим вимогам, у тому числі обмеження на отримання результатів, таких як час, гроші і ресурси.

Управління проектом – область діяльності, в ході якої визначаються та досягаються чіткі цілі проекту при балансуванні між обсягом робіт, ресурсами (такими як гроші, праця, матеріали, енергія, простір та ін.), часом, якістю та ризиками. Ключовим фактором успіху проектного управління є наявність чіткого заздалегідь визначеного плану, мінімізації ризиків і відхилень від плану, ефективного управління змінами [25][1)].

4.1 Планування проекту

Для поставленої задачі планування використовувалася програма MS Project 2010. Microsoft Project (або MSP) – програма керування проектами, розроблена компанією Microsoft.

Microsoft Project створений, щоб допомогти менеджеру проекту в розробці планів, розподілі ресурсів за завданнями, відстеженні прогресу і аналізі обсягів робіт. Microsoft Project створює розклад критичного шляху. Розклади можуть бути складені з урахуванням використовуваних ресурсів. Ланцюжок візуалізується на діаграмі Ганта.

Проект дозволяє досягти певного результату в певні терміни і за певні гроші. План проекту складається для того, щоб визначити, за допомогою яких робіт буде досягатися результат проекту, які люди й устаткування

---

1)1) [25] Реферат - Управління проектом. URL : http://masters.donntu.org/2017/ief/-panfilkina/diss/indexu.htm. (дата звернення 06.02.2019).

потрібні для виконання цих робіт, в який час ці люди й устаткування будуть зайняті роботою по проекту. Тому проектний план містить три основні елементи: завдання (task), ресурси (resource) і призначення (assignment).

У табл. 4.1 наведений план проекту, відповідно до якого велася його розробка і реалізація.

Дана таблиця містить всі завдання, які виконувалися при розробці інтерактивної моделі, тривалість кожної з задач, терміни її виконання, а також які ресурси для цього використовувалися.

Таблиця 1 – План проекту

Назва задачі	Термін	Початок	Кінець
1	2	3	4
Початок реалізації проекту	0 днів	28.10.19	28.10.19
Проектування системи	2 дні	28.10.19	30.10.19
Ознайомлення з предметною областю	2 дні	31.10.19	02.11.19
Постановка задачі	1 днів	02.11.19	03.11.19
Аналіз предметної області	2 дні	03.11.19	05.11.19
Інші роботи	3 дні	05.11.19	08.11.19
Проектування закінчено	0 днів	08.11.19	08.11.19

Продовження таблиці 4.1

1	2	3	4
Аналіз доступних середовищ розробки	2 дні	08.11.19	10.11.19
Вибір середовищ розробки	1 днів	10.11.19	11.11.19
Вивчення основ та характеристик Unity 3D	2 дні	11.11.19	13.11.19
Ознайомлення з анімацією у unity 3D	3 днів	13.11.19	16.11.19
Створення об’єктів у adobe photoshop	2 дні	16.10.19	18.11.19
Експорт об’єктів у Unity	0 днів	18.11.19	18.11.19
Розробка сцен у Unity	9 днів	18.11.19	27.11.19
Розробка фізики у Unity	2 дні	27.11.19	29.11.19
Розробка анімації	5 дні	29.11.19	04.12.19
Інші роботи	2 дні	04.12.19	06.12.19
Реалізація закінчена	0 днів	06.12.19	06.12.19
Експорт додатка	1 днів	07.12.19	08.12.19
Кінець проекту	1 днів	08.12.19	09.12.19

4.2 Управління ризиками проекту

Причиною виникнення ризиків є невизначеності, існуючі в кожному проекті. Ризики можуть бути відомі – ті, які визначені, оцінені, для яких

можливе планування. Ризики невідомі – ті, які не ідентифіковані і не можуть бути передбачені. Хоча специфічні ризики й умови їх виникнення не визначені, менеджери проекту знають, виходячи з минулого досвіду, що більшу частину ризиків можна передбачити.

Реалізуючи проекти, що мають високий ступінь невизначеності в таких елементах, як цілі і технології їх досягнення багато компаній приділяють увагу розробці та застосуванню корпоративних методів управління ризиками. Дані методи враховують як специфіку проектів, так і корпоративних методів управління.

Управління ризиками – це процеси, пов'язані з ідентифікацією, аналізом ризиків та прийняттям рішень, які включають максимізацію позитивних і мінімізацію негативних наслідків настання ризикових подій. Процес управління ризиками проекту зазвичай включає виконання наступних процедур:

- планування управлінням ризиками – вибір підходів та планування діяльності по керуванню ризиками проекту;
- ідентифікація ризиків – визначення ризиків, здатних впливати на проект і документування їх характеристик;
- якісна оцінка ризиків – якісний аналіз ризиків і умов їх виникнення з метою визначення їх впливу на успіх проекту;
- кількісна оцінка – кількісний аналіз ймовірності виникнення і впливу наслідків ризиків на проект;
- планування реагування на ризики – визначення процедур і методів з послаблення негативних наслідків ризикових подій і використання можливих переваг;
- моніторинг і контроль ризиків – моніторинг ризиків, визначення ризиків, що залишаються, виконання плану керування ризиками проекту і оцінка ефективності дій з мінімізації ризиків.

Всі ці процедури взаємодіють одна з одною, а також з іншими

процедурами. Кожна процедура виконується, принаймні, один раз в кожному проекті. Незважаючи на те, що процедури розглядаються як дискретні елементи з чітко визначеними характеристиками, на практиці вони можуть частково збігатися і взаємодіяти

При розробці даного наукового проекту не здійснювалося серйозне управління ризиками, проте враховувалася можливість виникнення певних форс-мажорів, таких як хвороби виконавців, забуті роботи і т.п. Для цього в кожен етап була додана фіктивна задача з мінімальним пріоритетом, під назвою «інші роботи» для кожного ресурсу, що можна побачити в табл. 4.1. Після вирівнювання ресурсів ці завдання виявляються в кінці етапу. Тривалість цих завдань залежить від імовірності виникнення і ступеня впливу ризиків, вона залежить від способу визначення оцінок тривалості задач, здоров’я членів команди та інших факторів.

Тривалість «інших робіт» для кожного етапу розробки виставлялася приблизно до чверті довжини етапу. Такий метод обліку ризиків дозволяє, по-перше назвати терміни виконання проекту та його етапів, причому аргументовано і з високим ступенем достовірності, і по-друге оцінити зразкові трудовитрати по проекту.

4.3 Оцінка трудомісткості і швидкості розробки програмного забезпечення

На даному етапі необхідно визначити, скільки часу буде займати реалізація проекту і в які терміни необхідно вирішити ту чи іншу задачу. Це, в свою чергу, визначить, коли саме потрібно залучити ті чи інші ресурси і як буде контролюватися розподіл коштів проекту.

Варто відзначити, що в управлінні термінами пріоритетним є визначення взаємозв’язків задач проекту і способів їх реалізації; саме це і впливає в першу чергу на те, як вибудовуються завдання в календарному плані робіт.

Даний підхід має ряд переваг.

По-перше, робота повинна визначати графік, а не навпаки, тому що досить легко розробити календарний план, який на ділі буде розходитися з намірами.

По-друге, даний підхід повністю співпадає з інструментами управління проектами, такими як Microsoft Project.

По-третє, використовуючи даний підхід, можна одержати цілком реалістичний графік результатів, менш схильних до змін, на відміну від графіків робіт, які є більш гнучкими і визначаються іншими способами.

Таким чином, процеси управління термінами проекту включають в себе наступні фази:

- визначення складу операцій – виявлення всіх планових операцій, які необхідно здійснити. Це досягається за допомогою таких методів, як: декомпозиція, шаблони, планування методом хвилі що набігає, експертна оцінка і т.д.;
- визначення взаємозв’язків операцій включає ідентифікацію і документування логічних взаємозв’язків між плановими операціями. Завдання послідовності може бути виконане за допомогою програмного забезпечення для керування проектами або вручну;
- оцінка ресурсів операцій повинна визначати, які ресурси (людські, матеріальні і т.д.) будуть використовуватися і в якій кількості, та коли кожен з цих ресурсів буде доступний для виконання проектних рішень;
- оцінка тривалості операцій використовує інформацію про склад робіт планової операції, типах необхідних ресурсів, розрахунковій кількості ресурсів і календарях ресурсів з указівкою їх доступності;
- розробка розкладу здійснюється за допомогою визначення планових дат початку і кінця кожної з операцій на проекті. Розробка розкладу здійснюється безперервно по всьому проекті по

мірі виконання робіт, зміни плану керування проектом і виникненні або припиненні очікуваних ризиків або виявленні нових;

– керування розкладом інтегрує в собі усі вищенаведені фази.

Проект був повністю завершений у строки 9 грудня 2019 року, як і було заплановано. Загальна тривалість розробки і реалізації проекту склала 42 дні. Під час розробки були задіяні всі наявні ресурси.

ВИСНОВКИ

При виконанні даної роботи було спроектовано і розроблено ігровий додаток з використанням багатоплатформового движка Unity. Додаток було розроблено під платформу Android. Була вибрана ця операційна система, тому що у даний час вона є найбільш популярна та швидко розвивається.

Усі сцени проекту створювались за допомогою ігрового движка Unity 3D та складаються з 92 різних спрайтів, з котрих 42 являються основними моделями, а тобто головний персонаж, антагоністи, платформи, пастки, декори. Також було розроблено 12 анімацій та задано їм умови для переключення. За всю гру було зроблено 50 сцен.

У даній роботі були розглянуті аспекти розробки ігрових додатків. Движок Unity показав себе як зручний, простий в освоєнні і професійно виконаний інструмент для створення ігор.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

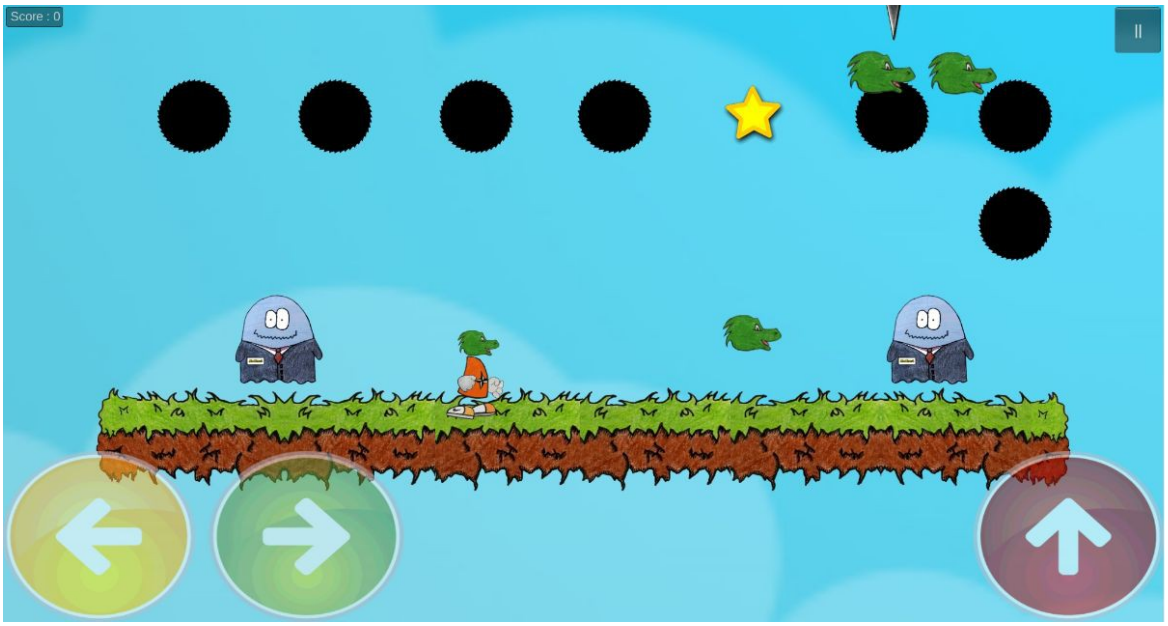
1. Програмні платформи сучасних смартфонів. URL : http://ukrefs.com.ua/ print:page,1,155256-Programmnye-platformy-sovre mennyh-smartfonov.html. (дата звернення 1.11.2019).
2. Яку вибрати платформу для створення мобільної гри | Каталог статей. URL : https://catalogueofarticles.com/uk/tehnologiyi/jaku-vibrati-platformudlja-stvorennja-mobilnoi-gri/ (дата звернення 02.11.2019).
3. List of game engines and frameworks. URL: https://en.wikipedia.org/wiki/ List_of_game_engines (дата звернення 08.11.2019).
4. DevGam сравнение игровых движков URL : http://devgam.com/sravne-nie- igrovyx-dvizhkov-kakoj-vybrat (дата звернення 08.11.2019).
5. Corona SDK – Википедия URL https://ru.wikipedia.org/wiki/Corona_SDK. (дата звернення 10.11.2019).
6. Unreal Engine – Википедия. URL https://ru.wikipedia.org/wiki/ Unre-al_Engine. (дата звернення 10.11.2019).
7. Phaser (game framework) – Wikipedia URL: https://en.wikipedia.org/ wiki/ Phaser_(game_framework) . (дата звернення 10.11.2019).
8. Cocos2d – Википедия URL: https://ru.wikipedia.org/wiki/Cocos2d . (дата звернення 11.11.2019).
9. Создание игры на движке Sprite Kit (Часть 1) / Хабр URL: https://habr.com/ru/post/225517/. (дата звернення 11.11.2019).
10. CreateJS – Wikipedia URL: https://en.wikipedia.org/wiki/CreateJS (дата звернення 11.11.2019).
11. AndEngine – Wikipedia URL: https://en.wikipedia.org/wiki/AndEngine (дата звернення 11.11.2019).
12. LibGDX – Википедия. URL: https://ru.wikipedia.org/wiki/LibGDX. (дата звернення 11.11.2019).

13. Unity (игровой движок) – Википедия. URL : https://ru.wikipedia.org/wiki/Unity_(игровой_движок). (дата звернення 11.11.2019).
14. Руководство Unity URL : https://docs.unity3d.com/ru/current/Manual/ index.html (дата звернення 12.11.2019).
15. Unity - Руководство: Audio Source. URL https://docs.unity3d.com/ru/current/Manual/class-AudioSource.html. (дата звернення 12.11.2019).
16. Unity - Руководство: Изучение интерфейса URL https://docs.unity3d.com/ru/current/Manual/LearningtheInterface.html. (дата звернення 12.11.2019).
17. Движок Unity – особенности, преимущества и недостатки. URL : http://gamesmaker.ru/3d-game-engines/unity3d/osobennostibesplatnogo-krossplatformennogo-dvizhka-unity3d/. (дата звернення 12.11.2019).
18. Оптимизация Unity UI / Блог компании FunCorp / Хабр. URL : https://habr.com/ru/company/funcorp/blog/470608/ (дата звернення 12.11.2019).
19. Форум для разработчиков игр. URL: http://www.gamedev.ru/forum/ (дата звернення 25.11.2019).
20. Unity - Руководство: Sprite Creator URL: https://docs.unity3d.com/ru/current/Manual/SpriteCreator.html (дата звернення 25.11.2019).
21. 2d анимация в unity - Your Game Now. URL : https://3dgame-creator.ru/unity-5-2d-animaciya/. (дата звернення 30.11.2019).
22. Unity - Руководство: Rigidbody 2D (Двухмерное твердое тело). URL : https://docs.unity3d.com/ru/current/Manual/class-Rigidbody2D.html. (дата звернення 02.12.2019).
23. Unity - Руководство: Коллайдеры (Colliders) https://docs.unity3d.com/ru/current/Manual/CollidersOverview.html (дата звернення 04.02.2019).
24. Unity - Руководство: Настройка Android SDK: https://docs.unity3d.com/ru/current/Manual/android-sdksetup.html. (дата звернення 04.02.2019).
25. Реферат - Управління проектом. URL : http://masters.donntu.org/2017/ief/panfilkina/diss/indexu.htm. (дата звернення 06.02.2019).

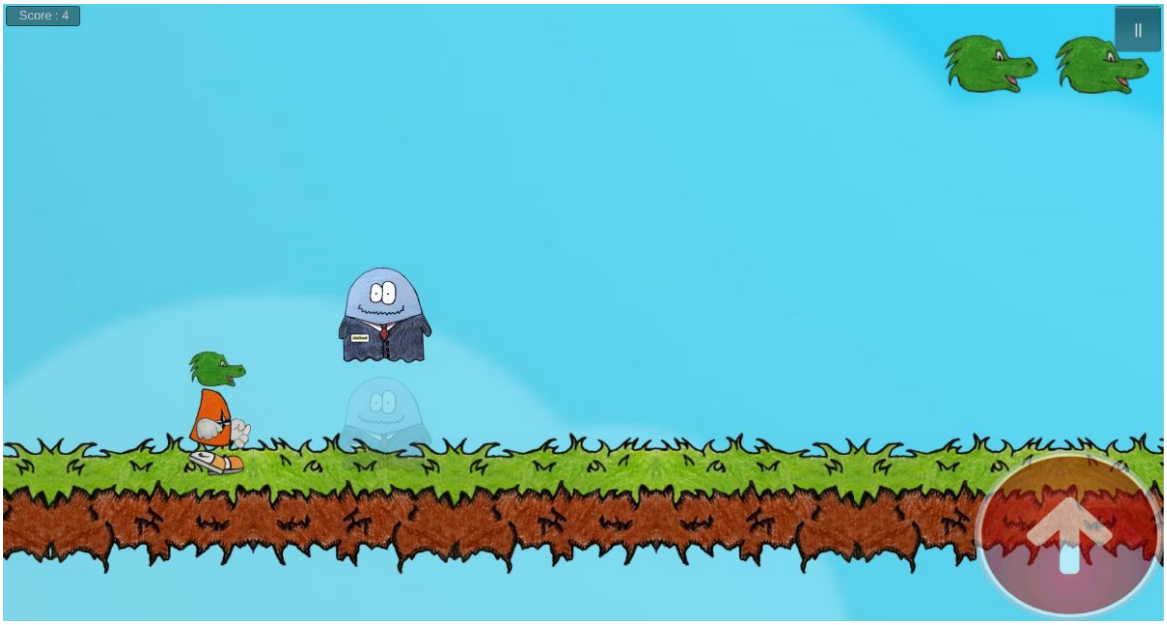
ДОДАТОК А

Презентація роботи мобільного додатку dino’s adventure

А.1 Інтерфейс гри Catch



А.2 Інтерфейс гри Run



А.3 Інтерфейс гри Adventures