Міністерство освіти та науки України

Національний технічний університет

«Харківський політехний інститут»

Факультет Комп’ютерних та інформаційних технологій

Кафедра «Системи інформацій»

Практична робота

Дисципліна «Технології розробки ігрових додатків»

Тема «Android гра “Flappy Bird”»

Виконала ст. гр.. КІТ218а

Скобеєва М. О.

Перевірив: Бреславець В. С.

Харків 2021

Зміст

[Вступ 3](#_Toc89985825)

[Етапи розробки типової гри 6](#_Toc89985826)

[Короткі відомості про гру “Flappy Bird” 8](#_Toc89985827)

[Підготовка до виробництва 9](#_Toc89985828)

[Виробництво 11](#_Toc89985829)

[Тестування 17](#_Toc89985830)

[Список використаних джерел 19](#_Toc89985831)

# **Вступ**

Розробка комп'ютерних ігор — процес створення ігор.

Розробкою комп'ютерних ігор може займатися як одна людина, і фірма (колектив розробників). Комерційні ігри створюються командами розробників, найнятими однією фірмою. Фірми можуть спеціалізуватись на виробництві ігор для персональних комп'ютерів, ігрових приставок або планшетних комп'ютерів. Розробка може фінансуватися іншою, більшою фірмою — видавцем. Фірма-видавець по закінченні розробки займається поширенням гри та бере на себе пов'язані з цим витрати. Протилежним підходом є така розробка, коли фірма самостійно (без участі видавців) розповсюджує копії ігор, наприклад, засобами цифрової дистрибуції.

Розробка найбільш великобюджетних ігор може коштувати десятки мільйонів доларів США, причому останні 2 десятиліття ці бюджети безупинно зростали, як і чисельність команд розробників і терміни розробки. Так, наприкінці дев'яностих гру для консолі PlayStation для кінцевого покупця — могла зробити команда з 10 осіб на рік, для PlayStation 2 (перша половина 2000-х років) потрібна була команда з 30-50 осіб і два роки розробки, до 2012 року мова йшлося вже про команди з понад 100 розробників і термін близько трьох років. За твердженням Алекса Мура, геймдизайнера з компанії Sumo Digital, якби ціна гри для кінцевого споживача зростала в тій же пропорції, ігри в 2012 коштували б по 1800 доларів США; інакше кажучи, щоб окупити зрослі бюджети за збереження тих самих цін магазинах, компанії-видавці мають продавати набагато більше копій игр.

Крупнобюджетна гра для двох платформ - Xbox 360 і PlayStation 3 - обходилася в 2012 в середньому в 20 мільйонів доларів, і для того, щоб вона окупилася, потрібно було продати близько двох мільйонів копій.

Unity (unity в перекладі з англ.  - «єдність») - міжплатформне середовище розробки комп'ютерних ігор, розроблене американською компанією Unity Technologies. Unity дозволяє створювати програми, що працюють на більш ніж 25 різних платформах, що включають персональні комп'ютери, ігрові консолі, мобільні пристрої, інтернет-програми та інші. Випуск Unity відбувся у 2005 році і з того часу триває постійний розвиток.

Основними перевагами Unity є наявність візуального середовища розробки, міжплатформної підтримки та модульної системи компонентів. До недоліків відносять появу складнощів при роботі з багатокомпонентними схемами та утруднення при підключенні зовнішніх бібліотек .

На Unity написані тисячі ігор, програм, візуалізації математичних моделей, які охоплюють безліч платформ і жанрів. У цьому Unity використовується як великими розробниками, і незалежними студіями.

Достоїнства і недоліки

Як правило, ігровий двигун надає безліч функціональних можливостей, що дозволяють їх задіяти в різних іграх, в які входять моделювання фізичних середовищ, карти нормалей, динамічні тіні та багато іншого. На відміну від багатьох ігрових движків, Unity має дві основні переваги: ​​наявність візуального середовища розробки і міжплатформенну підтримку. Перший фактор включає не тільки інструментарій візуального моделювання, а й інтегроване середовище, ланцюжок складання, що спрямоване на підвищення продуктивності розробників, зокрема етапів створення прототипів та тестування. Під міжплатформною підтримкою надається як місця розгортання (установка персональному комп'ютері, на мобільному пристрої, консолі тощо.), а й наявність інструментарію розробки (інтегроване середовище можна використовувати під Windows і Mac OS).

Третьою перевагою називається модульна система компонентів Unity, за допомогою якої відбувається конструювання ігрових об'єктів, коли останні є комбінованими пакетами функціональних елементів. На відміну від механізмів успадкування, об'єкти в Unity створюються за допомогою об'єднання функціональних блоків, а не поміщення у вузли дерева успадкування. Такий підхід полегшує створення прототипів, що є актуальним при розробці игр.

Як недоліки наводяться обмеження візуального редактора під час роботи з багатокомпонентними схемами, як у складних сценах візуальна робота не може. Другим недоліком називається відсутність підтримки Unity посилань на зовнішні бібліотеки, роботу з якими програмістам доводиться налаштовувати самостійно, і це ускладнює командну роботу. Ще один недолік пов'язаний із використанням шаблонів екземплярів (англ. prefabs). З одного боку, ця концепція Unity пропонує гнучкий підхід візуального редагування об'єктів, але з іншого боку, редагування таких шаблонів є складним. Також, WebGL-версія движка, в силу специфіки своєї архітектури (трансляція коду з C # С++ і далі в JavaScript), має ряд невирішених проблем з продуктивністю, споживанням пам'яті і працездатністю на мобільних пристроях.

**Етапи розробки типової гри**

1. Підготовка до виробництва

Підготовка до виробництва гри - це перший етап роботи над грою. Завдання розробників на цьому етапі - розробити концепцію гри, дизайн персонажів, вибрати засоби для реалізації проекту, створити прототип гри, підготувати план, за яким буде створюватися гра і узгодити цей план з начальством, або - з компанією, яка планує видавати гру. Як правило, всі сучасні ігри пишуться під конкретного видавця, який часто вкладає в розробку чималі кошти. Коли всі адміністративні питання вирішені, гра набуває на етап виробництва.

1. Виробництво

Виробництво - це ключовий етап у створенні гри. Розробники займаються реалізацією раніше створеного плану. Однак початковий план гри піддається змінам - іноді ці зміни відбуваються дуже часто - аж до щоденних коригувань. В ході виробництва гри - особливо це стосується комерційних версій - періодично влаштовується розгляд поточних результатів розробки, до яких команда повинна представити проект, який досяг певного рівня розвитку. Тобто, наприклад, до одного з таких моментів повинна бути готова працює демо-версія гри, до іншого - перший рівень і так далі. Як правило, ці проміжні результати служать відмінною рекламою нових ігрових проектів - демо-версії публікують на ігрових сайтах, геймери «приміряють» до цих версій можливості свого обладнання.

1. Випуск

Після того, як гра створена, протестована і налагоджена, настає час її випуску. Як правило, інтерес до цієї події посилено підігрівається видавцем гри - адже не варто забувати, що головна мета видавця - прибуток. Як правило, найбільш успішні гри з лишком виправдовують очікування видавців.

1. Підтримка

Ігри для ПК часто виходять з помилками - вся справа в тому, що розробникам вічно не вистачає часу щоб все як слід налагодити. Благо, є можливість виправляти помилки на вже встановлених іграх, встановлюючи патчі (від англійського patch - латка). Цим користуються розробники, випускаючи сируватий гру. Така практика не поширена для консольних ігор - тут розробники змушені відповідальніше підходити до своєї роботи і випускати повністю робочу гру, яка потребує втручань.

# **Короткі відомості про гру “Flappy Bird”**

Flappy Bird — гра для мобільних пристроїв, розроблена в'єтнамським розробником Донгом Нгуеном, в якій гравець за допомогою торкань екрана повинен контролювати політ птиці між рядами зелених труб, не зачіпаючи їх. Була реалізована на платформах iOS та Android.

Гра була випущена 24 травня 2013 року і набула раптового зростання популярності на початку 2014 року. Була розкритикована за складний ігровий процес, плагіат графіки та ігрової механіки, інші оглядачі вважали, що вона викликає залежність. Наприкінці січня 2014 стала безкоштовною грою в App Store. Протягом цього часу розробник заробляв $50 000 щодня за рекламу всередині докладання.

10 лютого 2014 року Flappy Bird була видалена розробником з App Store та Google Play. Після цього з'явилася тенденція продавати смартфони із заздалегідь встановленою грою, найчастіше за велику суму. Це також викликало появу безлічі клонів гри, створюваних розробниками, сподіваючись повторити успіх оригіналу.

Ігровий процес

Flappy Bird має ігровий процес за участю 2D-графіки. Мета гри полягає в управлінні польотом птаха, який безперервно пересувається між рядами зелених труб. При зіткненні із нею відбувається завершення гри. Управління проводиться торканням екрана, у якому птах робить невеликий ривок вгору. За відсутності ривків птах падає через силу тяжкості, і гра також завершується. Рахунок набираються при кожному успішному перельоті між двома трубами. Геймплей не має змін протягом усієї гри.

# **Підготовка до виробництва**

Для реалізації біла обрана гра під назвою “Flappy Bird”. Розробка буде здійснюватись на Unity з вікористанням 2D графіки під платформу Android.

Концепція гри полягає в управлінні польотом птаха, який безперервно пересувається між рядами зелених труб. При зіткненні із нею відбувається завершення гри. Управління проводиться торканням екрана, у якому птах робить невеликий ривок вгору. За відсутності ривків птах падає через силу тяжкості, і гра також завершується. Рахунок набираються при кожному успішному перельоті між двома трубами. Геймплей не має змін протягом усієї гри.

Головним персонажем гри була обрана пташка, як показано на рисунку 1.



Рисунок 1 – головний персонаж гри

Для фону гри була обрана картинка яка показана на рисунку 2.



Рисунок 2 – фон для гри

Також була створена земля (рисунок 3) на яку птах падає коли гра закінчується.



Рисунок 3 – земля

Були створені труби (рисунок 4) в якості перешкод.

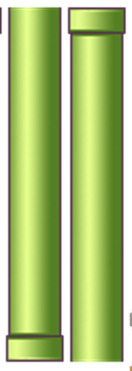


Рисунок 4 – труби (перешкоди)

Були використані елементи для упраління грою та результату проходження гри, такі як:

* Кнопка запуску гри (рисунок 5)
* Напис закінчення гри (рисунок 6)
* Напис з назвою гри (рисунок 7)
* Напис набраного рахунку (рисунок 8)



Рисунок 5 – кнопка запуску гри



Рисунок 6 – напис закінчення гри



Рисунок 7 – напис назви гри



Рисунок 8 – напис набраного рахунку

# **Виробництво**

Було створено дерево UI елементів гри (рисунок 9).

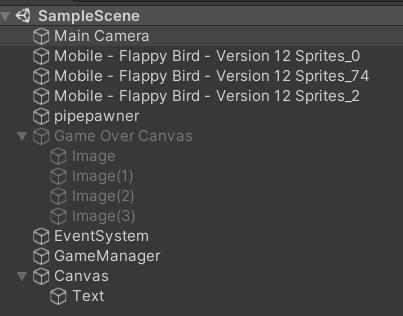


Рисунок 9 – дерево UI елементів гри

На головному персонажі гри були створені Rigidbody 2D та Capsule Collider 2D, для того щоб зчитувати зіткнення птички з перешкодами або землею (рисунок 10).

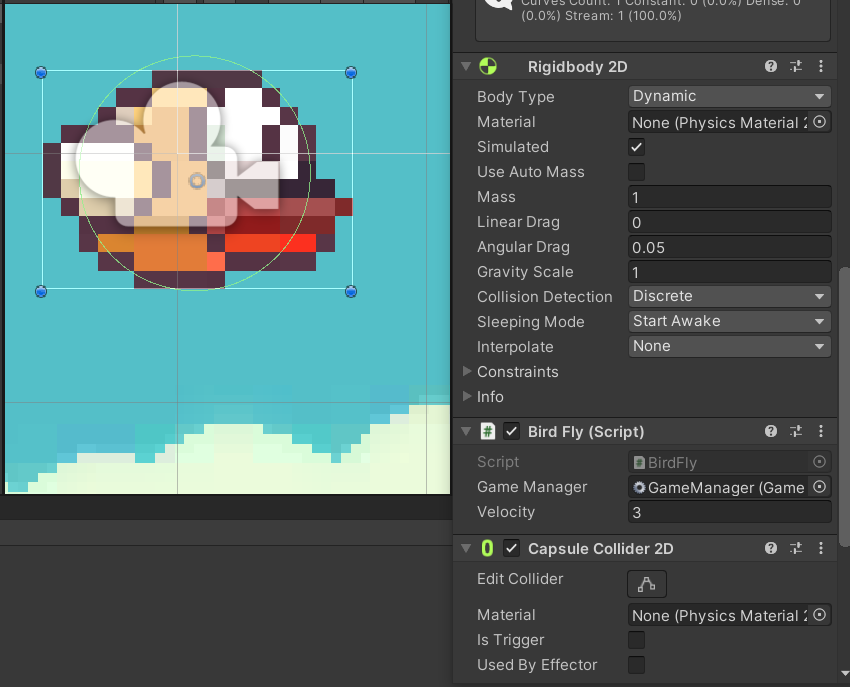


Рисунок 10 – створення компонентів для зчитування зіткнення птички и перешкодами

Був створений скрипт зі назвою BirdFly за допомогою якого птах робить невеликий ривок вгору.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class BirdFly : MonoBehaviour

{

public GameManager gameManager;

public float velocity = 1;

private Rigidbody2D rb;

void Start()

{

rb = GetComponent<Rigidbody2D>();

}

void Update()

{

if (Input.GetMouseButtonDown(0))

{

rb.velocity = Vector2.up \* velocity;

}

}

private void OnCollisionEnter2D(Collision2D collision)

{

gameManager.GameOver();

}

}

Був створений скрипт з назвою pipe для того щоб перешколи рухались.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class pipe : MonoBehaviour

{

public float speed; // скорость с которой двигаются трубы

void Start()

{

}

void Update()

{

transform.position += Vector3.left \* speed \* Time.deltaTime;

}

}

Був створений скрипт з назвою pipepawner який будет створювати перешкоди у випадкових місцях.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class pipepawner : MonoBehaviour

{

public float maxTime = 1;

private float timer = 0;

public GameObject pipe; // сюда добавятся рандомные трубы

public float height; // растояние между появляющимися трубами

void Start()

{

GameObject newpipe = Instantiate(pipe);

newpipe.transform.position = transform.position + new Vector3(0, Random.Range(-height, height), 0);

}

void Update()

{

if (timer > maxTime)

{

GameObject newpipe = Instantiate(pipe);

newpipe.transform.position = transform.position + new Vector3(0, Random.Range(-height, height), 0);

Destroy(newpipe, 15);

timer = 0;

}

timer += Time.deltaTime;

}

}

У Canvas додала елементи упаравління грою (кнопка початку гри, напис назви гри та закінчення гри). Також був створений об’єкт та скрипт GameManager, який відповідає за всю гру, а саме коли птах вризається у перешкоду або падає на землю гра запиняється и можно почати гру наново.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class GameManager : MonoBehaviour

{

public GameObject gameOverCanvas;

private void Start()

{

Time.timeScale = 1;

}

public void GameOver()

{

gameOverCanvas.SetActive(true);

Time.timeScale = 0;

}

public void Replay()

{

SceneManager.LoadScene(0);

}

}

До зображення кнопки почтаку гри додала компонент Button та образа функцію Replay із скрипту GameManager (рисунок 11).

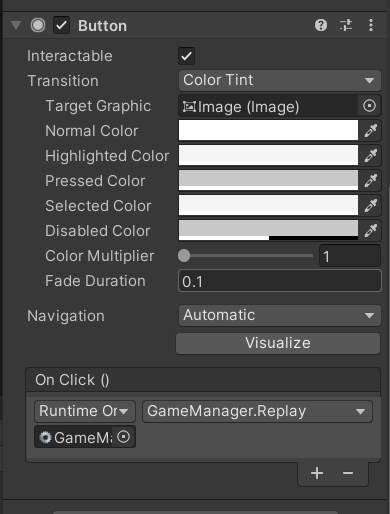


Рисунок 11 – додала функцію запуску гри на зображення кнопки

Створила скрипт з назвою Score для відображення набраного рахунку

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Score : MonoBehaviour

{

public static int score = 0;

private void Start()

{

score = 0;

}

private void Update()

{

GetComponent<UnityEngine.UI.Text>().text = score.ToString();

}

}

Створила компонент під назвою Box Collider 2D (рисунок 12) через який птиця пролетає та рахунок збільшується на 1 бал

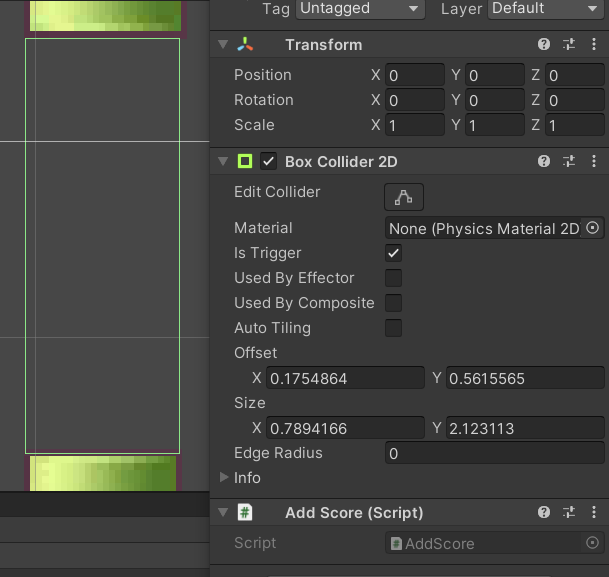


Рисунок 12 – Створеній компонент Box Collider 2D

Створила скрипт з назвою AddScore який збільшує набраних рахунок коли птица пролітає перешкодіни та не врізається в них.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class AddScore : MonoBehaviour

{

private void OnTriggerEnter2D(Collider2D collision)

{

Score.score++;

}

}

# **Тестування**

Початковий єкран гри (рисунок 13)

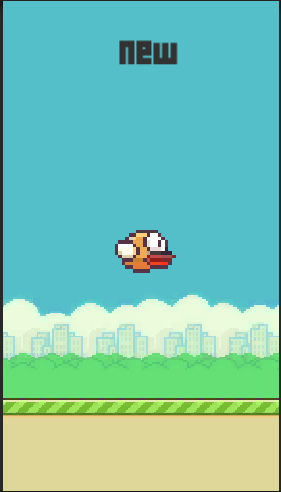


Рисунок 13 – початковий єкран гри

Єкран гри коли птиця врзається у перешкоду або падає на землю (рисунок 14)

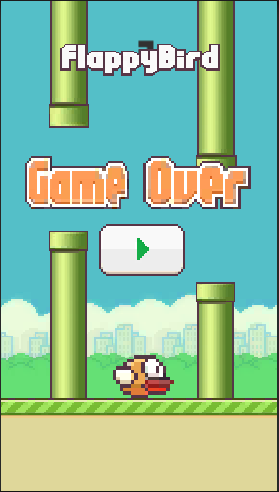


Рисунок 14 – Закінцення гри

Єкран гри під час проходження гри (рисунок 15)

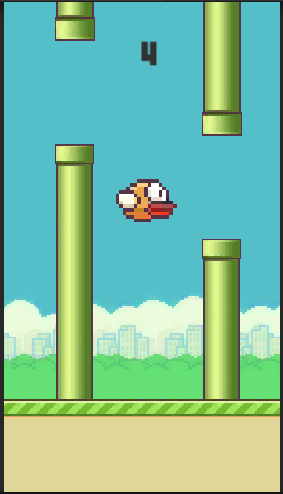


Рисунок 16 – проходження гри

Висновок з тестування: гра готова до випуску, ніяких серьезних помилок не знайдено, все процює як и планувалось.

**Список використаних джерел**

1. Спрайти для UI гри [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://drive.google.com/file/d/19yZeyrmbTm2dlnrUV4JytVx3YZZJXurA/view>.
2. Разработка компьютерных игр [Електронний ресурс] // Вікіпедія. – 2021. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D1%85_%D0%B8%D0%B3%D1%80>.
3. Unity (игровой движок) [Електронний ресурс] // Вікіпедія. – 2021. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Unity_(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA)>.
4. Flappy Bird [Електронний ресурс] // Вікіпедія. – 2021. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Flappy_Bird>.
5. Шрифт flappy bird [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://drive.google.com/drive/u/0/folders/12h5VoePCWBtLdo1woooPRk6aDLHHzeor>.