

**TRƯỜNG ĐẠI HỌC THÔNG TIN LIÊN LẠC  
KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN  
LẬP TRÌNH GAME VÀ MÔ PHỎNG**

**Đề Tài:  
Phát Triển Game Flappy Balloon trên Unity**

**GVHD: Nguyễn Việt Hùng**

**SVTH: Trần Trung Hiếu**

**Lê Huy Hoàng**

**Bùi Lý Hải Đăng**

**LỚP: ĐHCN1A**

**NHA TRANG, tháng 01 năm 2018**

TRƯỜNG ĐẠI HỌC THÔNG TIN LIÊN LẠC  
KHOA CÔNG NGHỆ THÔNG TIN  
LỚP ĐHCN1A

**ĐỀ TÀI:**  
**PHÁT TRIỂN GAME FLAPPY  
BALLOON  
TRÊN UNITY**



**GVHD:** Nguyễn Việt Hùng

**Nhóm sinh viên thực hiện:**

1. Trần Trung Hiếu ..... 14ĐC027 .... [tranhieukk@gmail.com](mailto:tranhieukk@gmail.com)..... 01677136466
2. Lê Huy Hoàng ..... 14ĐC127 .... pronbk96@gmail.com ..... 01283566999
3. Bùi Lý Hải Đăng ..... 14ĐC009 .... builyhaidang@gmail.com ..... 0964845216

*Nha Trang, tháng 1 năm 2018*

[illegible]

# LỜI MỞ ĐẦU

Trong thời đại công nghệ thông tin như hiện nay, sản phẩm công nghệ ngày càng chịu sự đánh giá khắt khe hơn từ phía những người dùng, đặc biệt là về sản phẩm Game được nhận rất nhiều sự đánh giá từ phía các Game thủ, hay chỉ là những người chơi bình thường. Ngành công nghiệp Game hiện nay có thể nói là bùng nổ, với tốc độ phát triển đến chóng mặt, rất nhiều những Game hay và hấp dẫn đã được ra đời trong thời gian qua. Phía sau những Game phát triển và nổi tiếng như vậy đều có một Game Engine.

Từ xu hướng phát triển và những bất cập trên, đồ án này sẽ khảo sát và nghiên cứu về Engine Unity – một Game Engine rất phổ biến và không kém mạnh mẽ hiện nay nhằm thực nghiệm việc phát triển một trò chơi (Demo) Flappy Balloon-game đã từng rất phát triển trên cộng đồng game di động.

Với đề tài và môn học này, nhóm xin chân thành cảm ơn sự giúp đỡ tận tình của giảng viên **Nguyễn Việt Hùng** - INFOdation. Song, do còn nhiều hạn chế đề tài nhóm xây dựng không tránh khỏi những thiếu sót. Rất mong được anh và các bạn đóng góp ý kiến để chương trình ngày càng hoàn thiện và được đưa vào sử dụng.

**Trân trọng cảm ơn!**

Nha Trang, tháng 1 năm 2018

*Nhóm thực hiện.*

# MỤC LỤC

<b>CHƯƠNG 1:CƠ SỞ LÝ THUYẾT .....</b>	<b>5</b>
1.1 Unity là gì? .....	5
1.2 Tính năng của engine Unity .....	6
<b>CHƯƠNG 2:KẾ HOẠCH THỰC HIỆN .....</b>	<b>8</b>
2.1.Yêu cầu thực hiện .....	8
2.2.Mục đích thực hiện .....	8
<b>CHƯƠNG 3:XÂY DỰNG GIAO DIỆN,VIẾT MÃ LỆNH GAME.....</b>	<b>9</b>
3.1.Ý tưởng.....	9
3.2.Xây dựng nhân vật .....	9
3.3.Viết mã lệnh cho game.....	9
3.3.1.Nhân vật.....	9
3.3.2.Quản lý game .....	14
<b>CHƯƠNG 4:BIÊN DỊCH VÀ KIỂM THỬ' .....</b>	<b>16</b>
4.1 Biên dịch.....	16
<b>CHƯƠNG 5: KẾT LUẬN .....</b>	<b>19</b>
5.1.Uưu điểm .....	19
5.2.Hạn chế.....	19
5.3. Các kỹ năng học được.....	19
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>20</b>

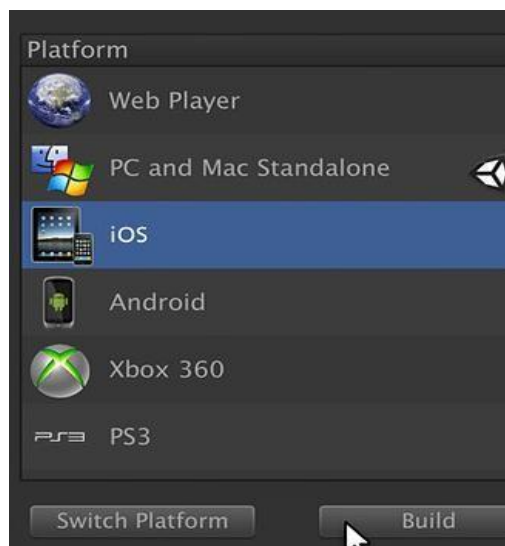
# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1.1 Unity là gì?

Đã qua rồi thời kỳ làm game trên nền Flash căn bản và buồn chán với những chuyển động thật cứng nhắc. Unity mang lại sức mạnh kỳ diệu cho nhân vật mà chúng ta muốn thể hiện sống động hơn trong không gian 3 chiều đầy huyền ảo. Công nghệ cao này tạo ra một bước đột phá mới về sự khác biệt trong công nghệ làm game hiện nay, mang đến cho người chơi 1 cảm giác rất khác lạ và hào hứng trong từng chuyển động, tương lai công nghệ này được áp dụng vào game Việt Nam sẽ mở ra một trang mới trong thế giới game 2D, 3D huyền ảo.

Unity được dùng để làm video game, hoặc những nội dung có tính tương tác như thể hiện kiến trúc, hoạt hình 2D, 3D thời gian thực. Unity hao hao với Director, Blender game engine, Virtools hay Torque Game Builder trong khía cạnh dùng môi trường đồ họa tích hợp ở quá trình phát triển game là chính.

Unity là một trong những engine được giới làm game không chuyên cực kỳ ưa chuộng bởi khả năng tuyệt vời của nó là phát triển trò chơi đa nền. Trình biên tập có thể chạy trên Windows và Mac OS, và có thể xuất ra game cho Windows, Mac, Wii, iOS, Android. Game cũng có thể chơi trên trình duyệt web thông qua plugin Unity Web Player. Unity mới bổ sung khả năng xuất ra game trên widget cho Mac, và cả Xbox 360, PlayStation 3.



## **Hình 1.1 Hình minh họa đa nền**

Chỉ với khoản tiền bỏ ra khá khiêm tốn (1.500 USD) là phiên bản pro đã nằm trong tay của chúng ta, dĩ nhiên tại Việt Nam số tiền này vẫn là quá lớn nhưng thật may là đã có phiên bản Unity Free. Tuy nhiên, nhiều tính năng quan trọng (Network) bị cắt giảm nhưng đó không phải là vấn đề quá lớn nếu muốn phát triển một tựa game tầm trung.

Vào năm 2009, Unity nằm trong top 5 game engine tốt nhất cho việc sản xuất game với chỉ sau 4 năm phát triển. Unity đứng thứ 4, xếp sau Unreal Engine 3, Gamebryo Engine (được VTC Studio mua về phát triển SQUAD) và Cry Engine 2. Lượng tài liệu hướng dẫn Unity rất phong phú. Hơn thế nữa nó còn có sẵn một cộng đồng cực lớn với diễn đàn riêng. Bất cứ điều gì không hiểu chúng ta đều có thể thoải mái hỏi và nhận được câu trả lời nhanh chóng, tận tâm.

Quá trình tạo địa hình cũng như truy xuất từ các phần mềm 3DSMax, Maya, Cinema4D... rất nhanh chóng. Sức mạnh và sự tiện lợi của Unity là vô cùng lớn.

❖ **Sức mạnh:** Unity có thể tạo ra được nhiều loại game 2D, 3D đa dạng, dễ sử dụng với người làm game chưa chuyên nghiệp, chất lượng cao, chạy hầu hết trên các hệ điều hành.

❖ **Sự tiện lợi:** nếu chúng ta là một người chuyên dùng 3Dmax, hay Maya hoặc phần mềm mã nguồn mở Blender thì quả là thật tuyệt, chúng ta sẽ có một lợi thế lớn khi viết game trên Unity này, bởi công việc tạo các mô hình 2D, 3D sẽ trở lên dễ dàng hơn rất nhiều, việc kết hợp giữa người lập trình và người thiết kế các mô hình sẽ nhanh và hiệu quả hơn. Trong Unity chúng ta có thể import trực tiếp các file mô hình đang thiết kế và sẽ thiết kế hoàn thiện tiếp nếu chưa xong trong khi đó công việc import chỉ diễn ra một lần. Không như việc phải dùng các công cụ khác để thực hiện viết game chúng ta sẽ phải xuất chúng ra một dạng nào đó và mỗi lần sửa lại phần mô hình chúng ta lại phải import lại, và như thế là quá mất thời gian trong việc tạo và chỉnh sửa các mô hình theo ý muốn. Ngoài ra Unity còn cho chúng ta trực tiếp tạo các mô hình nếu muốn. Việc đặt các thuộc tính vật lý trong Unity cũng cực kỳ dễ dàng và hỗ trợ sẵn nhiều chức năng.

## **1.2 Tính năng của engine Unity**

❖ **Môi trường phát triển** được tích hợp với tính năng kế thừa, khả năng chỉnh sửa

đồ họa, chức năng kiểm tra chi tiết, và đặc biệt tính năng xem trước game ngay trong lúc xây dựng (live game preview).

❖ Triển khai được trên nhiều nền tảng:

- Chương trình độc lập trên Windows và Mac OS.
- Trên web, thông qua Unity Web Player plugin cho Internet Explorer, Firefox, Safari, Opera, Chrome, cho cả Windows và Mac OS.
- Trên Mac OS Dashboard widget.
- Cho Nintendo Wii (cần mua license thêm.)
- Cho iPhone, iPad application (cần mua license thêm.)
- Cho Google Android (cần mua license thêm.)
- Cho Microsoft Xbox 360 (cần mua license thêm.)
- Cho Sony PlayStation 3 (cần mua license thêm.)

❖ Tài nguyên (model, âm thanh, hình ảnh, ...) được tải vào trong Unity và tự động cập nhật nếu tài nguyên có sự thay đổi. Unity hỗ trợ các kiểu định dạng từ 3DS Max, Maya, Blender, Cinema 4D và Cheetah3D.

❖ Graphics engine sử dụng Direct3D (Windows), OpenGL (Mac, Windows), OpenGL ES (iPhone OS), và các API khác trên Wii.

❖ Hỗ trợ bump mapping, reflection mapping, parallax mapping, Screen Space Ambient Occlusion v...v...

❖ Unity Asset Server: Đây là một tính năng khá mới của Unity, theo đó Unity sẽ cung cấp một hệ thống quản lý theo dạng phiên bản cho tất cả asset và cả script. Đây là một kho chứa các tài nguyên cần thiết cho việc làm game. Khi import cũng như sửa chữa, trạng thái của asset ngay lập tức được cập nhật. Server chạy trên database opensource PostgreSQL và có thể truy cập trên cả Mac lẫn Windows, Linux. Asset Server đòi hỏi một khoản phí phụ trội là \$499 cho mỗi bản copy Unity, và một license Unity Pro.



## CHƯƠNG 2: KẾ HOẠCH THỰC HIỆN

### 2.1. Yêu cầu thực hiện

1. Xây dựng khung chương trình và các quang cảnh cần thiết trong trò chơi.
2. Xây dựng các đối tượng cần thiết trong trò chơi.
3. Xây dựng các mức độ trò chơi khi người dùng chơi với máy tính và nhiều người cùng chơi với nhau.
4. Xây dựng các setting cần thiết để người chơi điều chỉnh

### 2.2. Mục đích thực hiện

Xây dựng game một trò chơi mô phỏng chạy trên nền tảng Unity, biên dịch ra nền tảng di động .

#### Tên game: “Flappy Balloon”

Thời gian thực hiện dự án : 25/12/2017-18/1/2018

Bảng phân công chi tiết

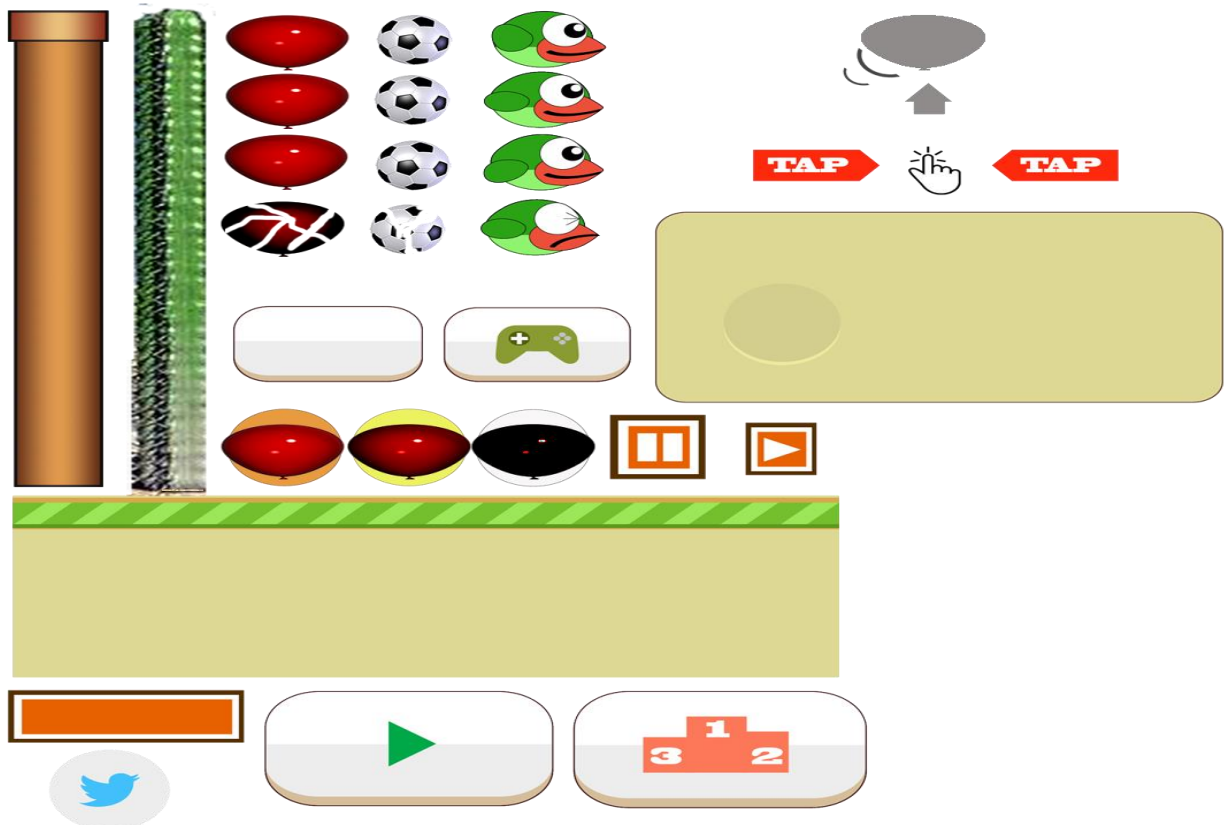
Stt	Công việc	Thời gian thực hiện	Người thực hiện	Ghi chú
01	Lên ý tưởng, tìm hiểu cách dùng Unity	25/01/2017-1/1/2018	Hiếu Hoàng Đăng	
02	Xây dựng nhân vật, viết code tương tác game	1/1/2018-8/1/2018	Hoàng Hiếu	
03	Xây dựng giao diện, sửa lỗi game	8-10/01/2018	Đăng Hiếu	
04	Test game, viết báo cáo, làm slide thuyết trình	10-13/1/2018	Đăng Hoàng	
05	Chỉnh sửa và hoàn thiện game	13-18/1/2018	Hiếu Hoàng Đăng	

## CHƯƠNG 3:XÂY DỰNG GIAO DIỆN,VIẾT MÃ LỆNH GAME

### 3.1.Ý tưởng

Xây dựng hình tượng bóng bay bay đi qua các xương rồng.Với mỗi lần bay qua hai xương rồng sẽ được cộng một điểm,nếu chạm vào xương rồng sẽ bị chết.Lần chơi qua được nhiều nhất sẽ báo điểm cao nhất,hệ thống sẽ so sánh các lần chơi với điểm cuối cùng để đưa ra điểm cao nhất.

### 3.2.Xây dựng nhân vật



Hình 3.1.Nhân vật trong game

### 3.3.Viết mã lệnh cho game

#### 3.3.1.Nhân vật

##### 3.3.1.1.Background

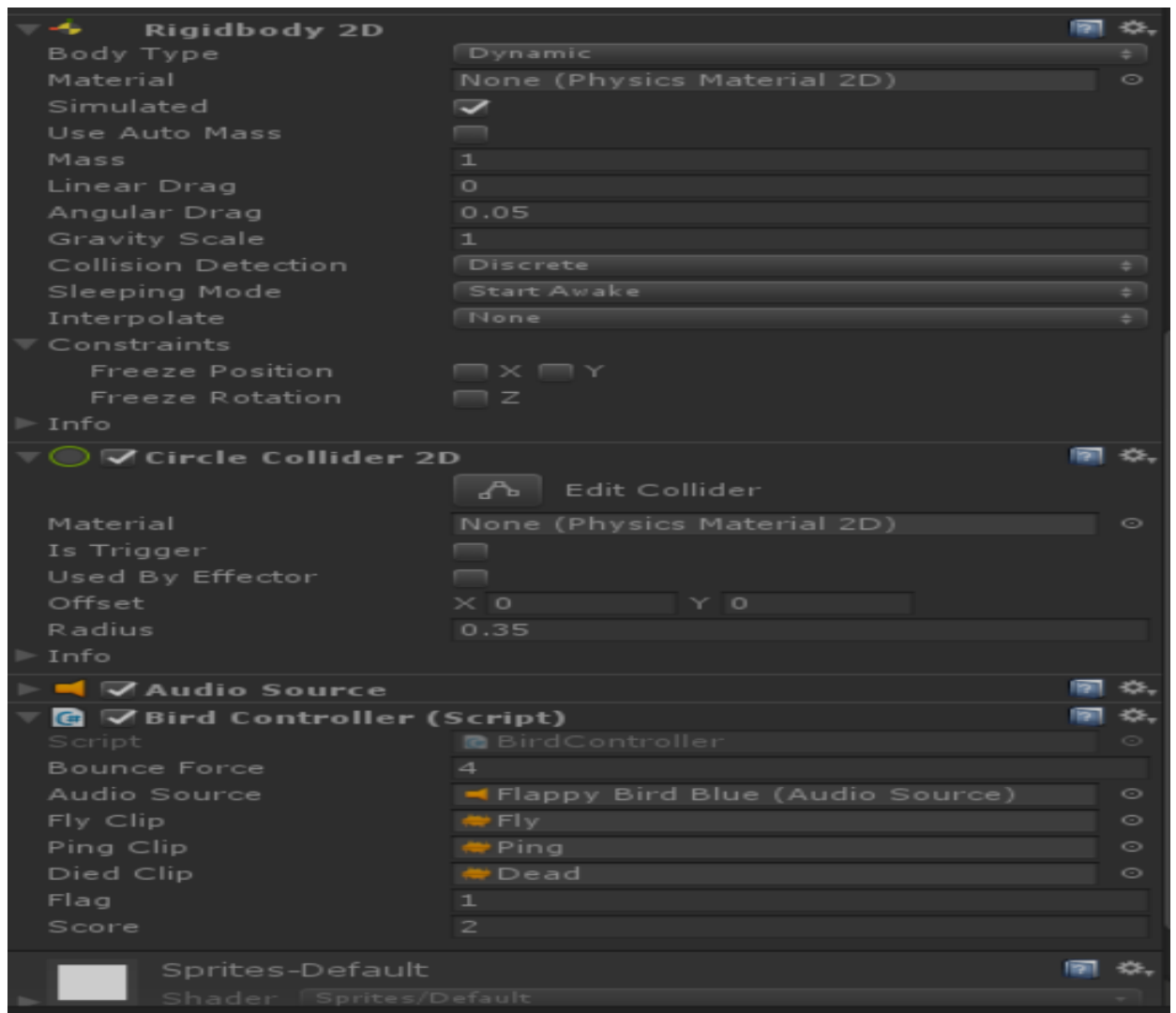


**Hình 3.2 Background game**

Sử dụng độ phân giải 480x800 dùng làm background chính hiển thị trong game. Do không có sự đồng đều về độ phân giải khi biên dịch, do đó một đoạn script sẽ xử lý về độ phân giải màn hình để hiển thị tốt hơn trên màn hình điện thoại.

```
void Start () {  
    SpriteRenderer sr = GetComponent<SpriteRenderer> ();  
    Vector3 tempScale = transform.localScale;  
    float height = sr.bounds.size.y;  
    float width = sr.bounds.size.x;  
    float worldHeight = Camera.main.orthographicSize * 2f; //10  
    float worldWidth = worldHeight * Screen.width / Screen.height; //10 *  
    203/339  
    tempScale.y = worldHeight / height;  
    tempScale.x = worldWidth / width;  
    transform.localScale = tempScale;  
}
```

### 3.3.1.2. Ball Animation



**Hình 3.3 Các thuộc tính của Ball Animation**

Có hai thành phần quan trọng là **Rigidbody 2D** và **Circle Collider 2D** xác định trọng lực của bóng và độ va chạm của bóng với cây xương rồng

**Rigidbody 2D:** Gravity Scale=1 ,biểu hiện có lực hút của trọng lực

**Circle Collider 2D:** Radius=0.35 xác định sự va chạm với cây xương rồng

**BirdController:**viết mã C# cho quả bóng

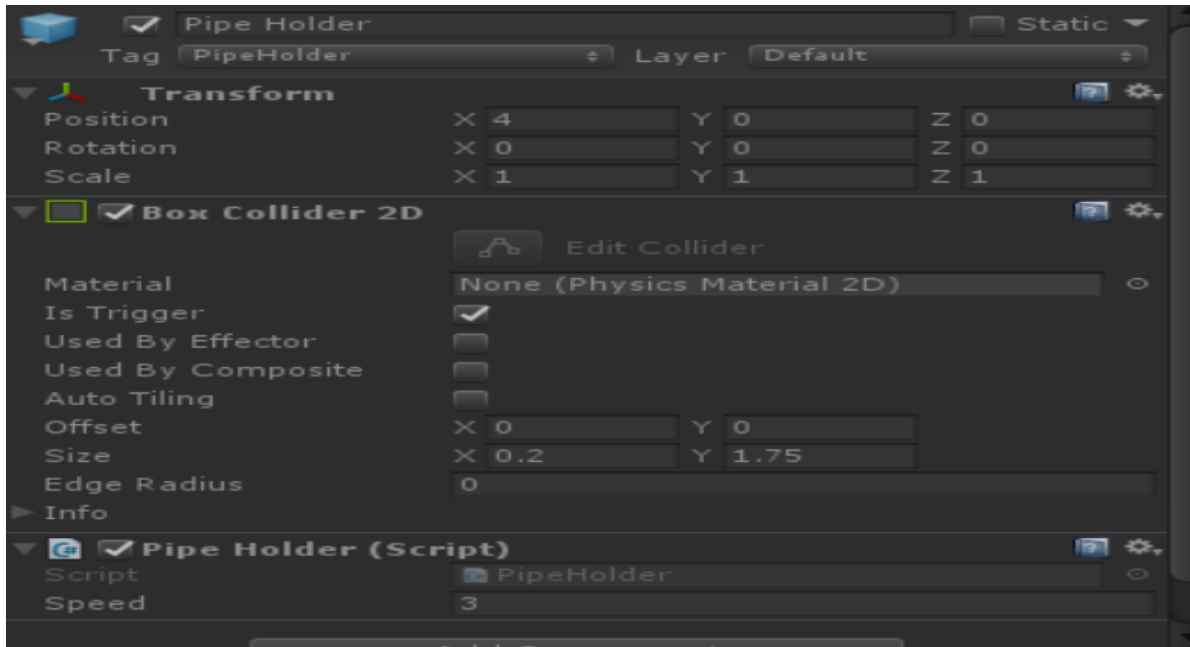
```

public class BirdController : MonoBehaviour { // Use this for initialization
    void Awake () {
        isAlive = true;    myBody = GetComponent<Rigidbody2D> ();
        anim = GetComponent<Animator> (); _MakeInstance ();
        spawner = GameObject.Find ("Spawner Pipe"); //Spawner Pipe :sinh cây tự động
    }
    void _MakeInstance(){if (instance == null) {instance = this;}}
    void FixedUpdate () {_BirdMoveMent ();}
    void _BirdMoveMent(){if (isAlive) {if (didFlap) {didFlap = false;
        myBody.velocity = new Vector2 (myBody.velocity.x, bounceForce);
        audioSource.PlayOneShot (flyClip);}}
        if (myBody.velocity.y > 0) {
            float angel = 0;
            angel = Mathf.Lerp (0, 90, myBody.velocity.y / 7);
            transform.rotation = Quaternion.Euler (0, 0, angel);
        }else if (myBody.velocity.y == 0) {
            transform.rotation = Quaternion.Euler (0, 0, 0);
        }else if (myBody.velocity.y < 0) {
            float angel = 0;
            angel = Mathf.Lerp (0, -90, -myBody.velocity.y / 7);
            transform.rotation = Quaternion.Euler (0, 0, angel);}}
    public void FlapButton(){didFlap = true;}
    void OnTriggerEnter2D(Collider2D target){
        if (target.tag == "PipeHolder") {
            score++;
            if (GamePlayController.instance != null) {
                GamePlayController.instance._SetScore (score);}
            audioSource.PlayOneShot (pingClip);}}

    void OnCollisionEnter2D(Collision2D target){
        if (target.gameObject.tag == "Pipe" || target.gameObject.tag == "Ground") {
            flag = 1; if (isAlive) {
                isAlive = false;
                Destroy (spawner);
                audioSource.PlayOneShot (diedClip);
                anim.SetTrigger ("Died");
            }
            if (GamePlayController.instance != null) {
                GamePlayController.instance._BirdDiedShowPanel
(score);
            }
        }
    }
}

```

### 3.3.1.3.Xương rồng



Hình 3.4 Các thuộc tính của cây xương rồng

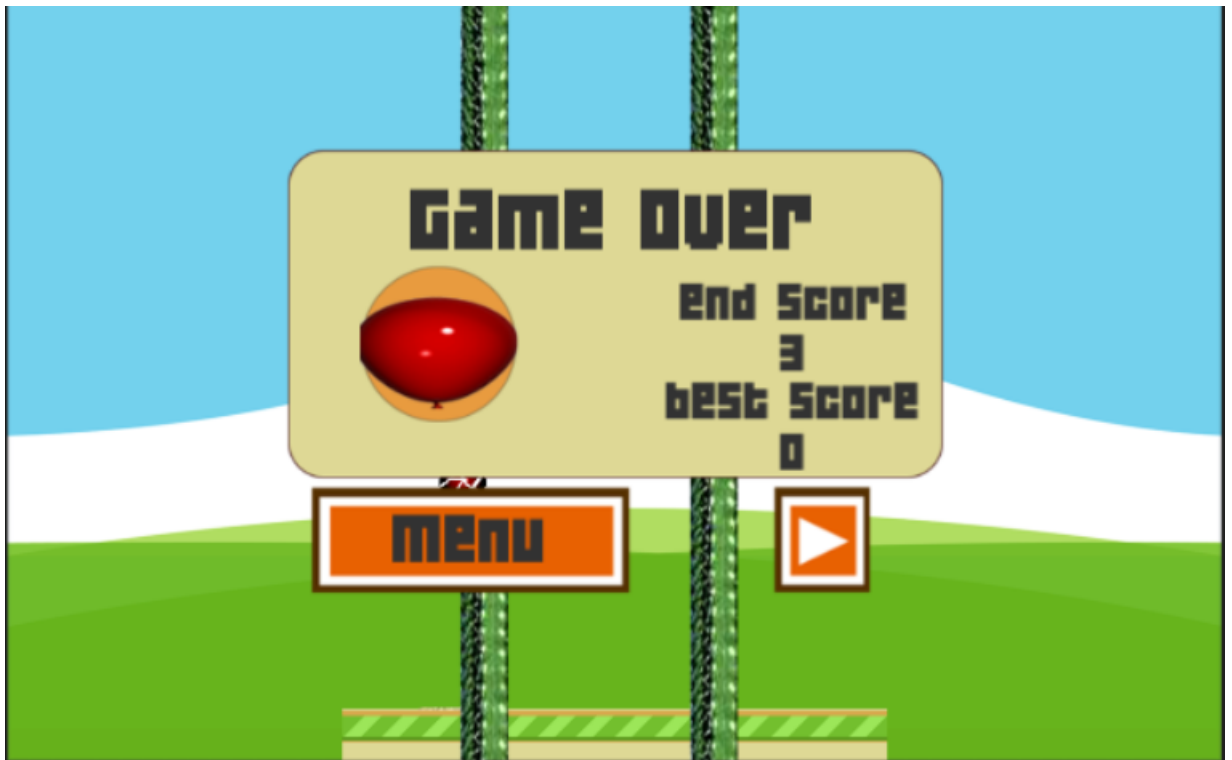
Xương rồng bao gồm hai ống:câytrên và câydưới

**Box Collider 2D** tương tự như **Circle Collider 2D** xác định độ va chạm dạng hình hộp

**Pipe Holder:**

```
// Update is called once per frame
void Update () {
    if (BirdController.instance != null) {
        if (BirdController.instance.flag == 1) {
            Destroy (GetComponent<PipeHolder> ());}}
    _PipeMovement ();}
void _PipeMovement(){
    Vector3 temp = transform.position; temp.x -= speed * Time.deltaTime;
    transform.position = temp;}
void OnTriggerEnter2D(Collider2D target){
    if (target.tag == "Destroy") {Destroy (gameObject);}}
```

Qua quá trình làm, nhận thấy các cây xương rồng không thể cứ tạo lên xuống cây theo một chiều nhất định,do đó tạo thêm một đoạn code Spanwer Pipe có tác dụng tạo ngẫu nhiên các xương rồng trong Spawner Pipe Script



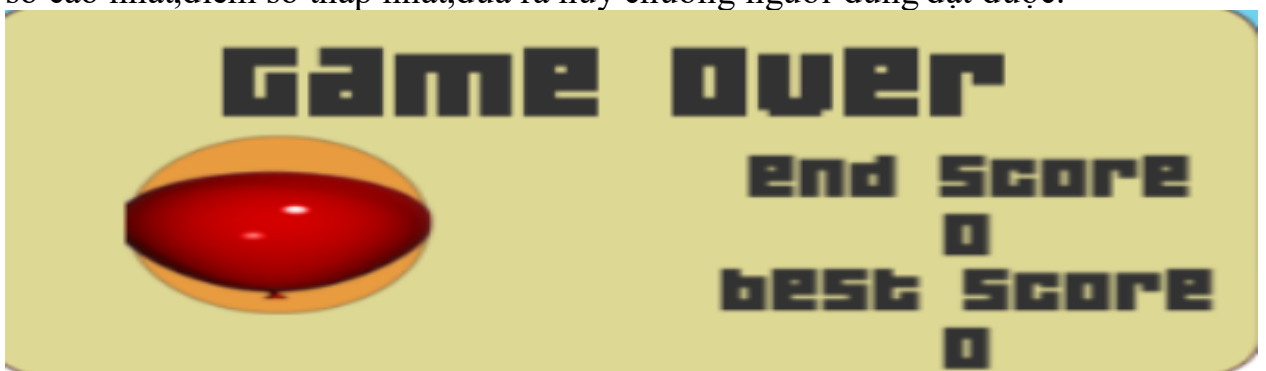
Hình 3.5 Tạo xương rỗng tự động

```
void Start () {
    StartCoroutine (Spawner ());
}

IEnumerator Spawner(){
    yield return new WaitForSeconds (1);
    Vector3 temp = pipeHolder.transform.position;
    temp.y = Random.Range (-2.5f, 2.5f);
    Instantiate (pipeHolder, temp, Quaternion.identity);
    StartCoroutine (Spawner ());
}
```

### 3.3.2.Quản lý game

Khi người chơi game sẽ có một bộ phận quản lý game cũng như đưa ra điểm số cao nhất, điểm số thấp nhất, đưa ra huy chương người dùng đạt được.



Hình 3.6.Kết quả game

```

void IsGameStartedForTheFirstTime() {
    if (!PlayerPrefs.HasKey ("IsGameStartedForTheFirstTime")) {
        PlayerPrefs.SetInt (HIGH_SCORE, 0);
        PlayerPrefs.SetInt ("IsGameStartedForTheFirstTime", 0);
    }
}

void _MakeSingleInstance(){if (instance != null) {
    Destroy (gameObject);
} else {
    instance = this;
    DontDestroyOnLoad (gameObject);
}
}

public void SetHighScore(int score){
    PlayerPrefs.SetInt (HIGH_SCORE, score);
}

public int GetHighScore(){
    return PlayerPrefs.GetInt (HIGH_SCORE);
}

public void _BirdDiedShowPanel(int score){
    gameOverPanel.SetActive (true);
    endScoreText.text = "" + score;
    if (score > GameManager.instance.GetHighScore ()) {
        GameManager.instance.SetHighScore (score);
    }
    bestScoreText.text = "" +GameManager.instance.GetHighScore();}

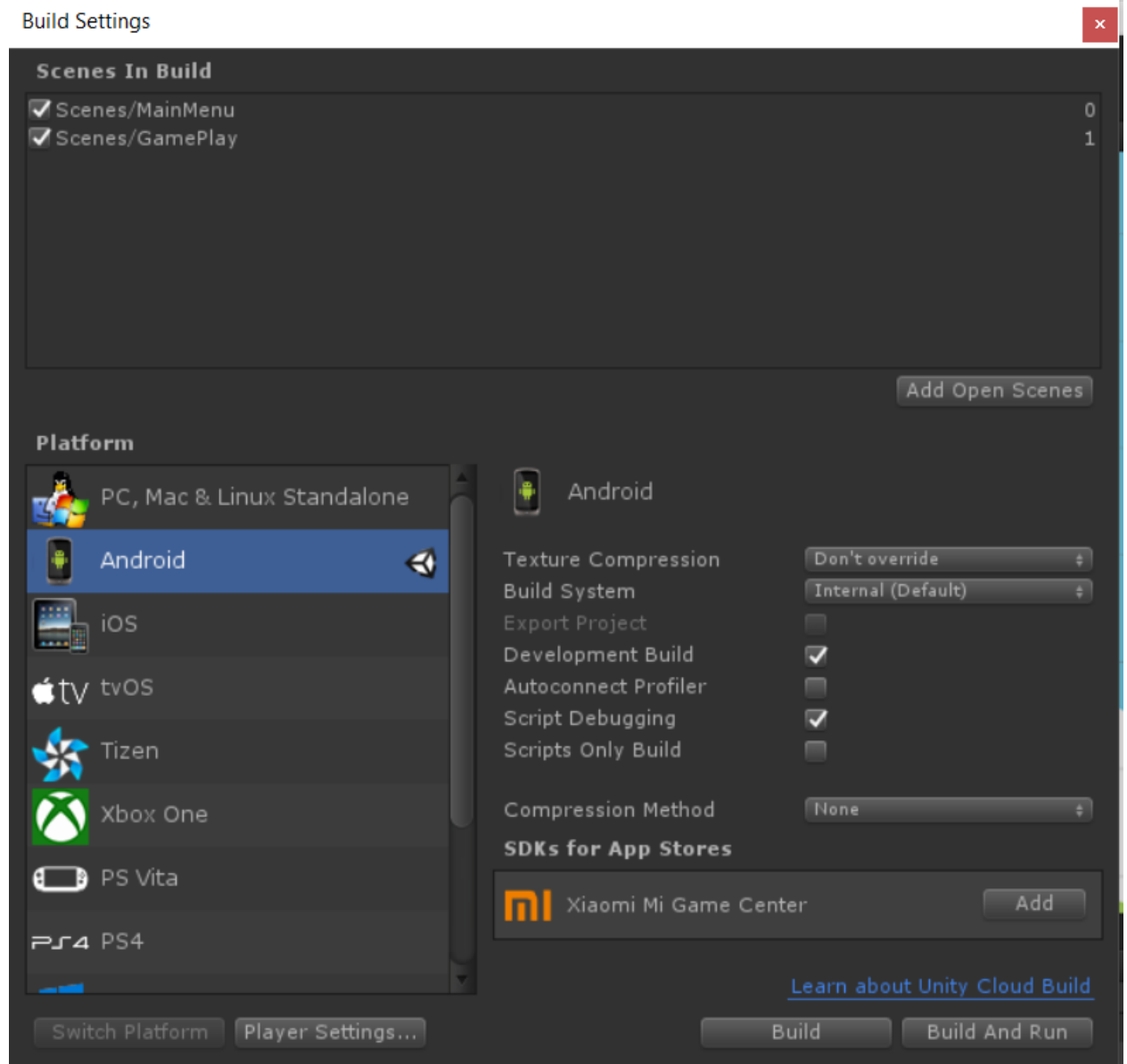
```



## CHƯƠNG 4: BIÊN DỊCH VÀ KIỂM THỬ

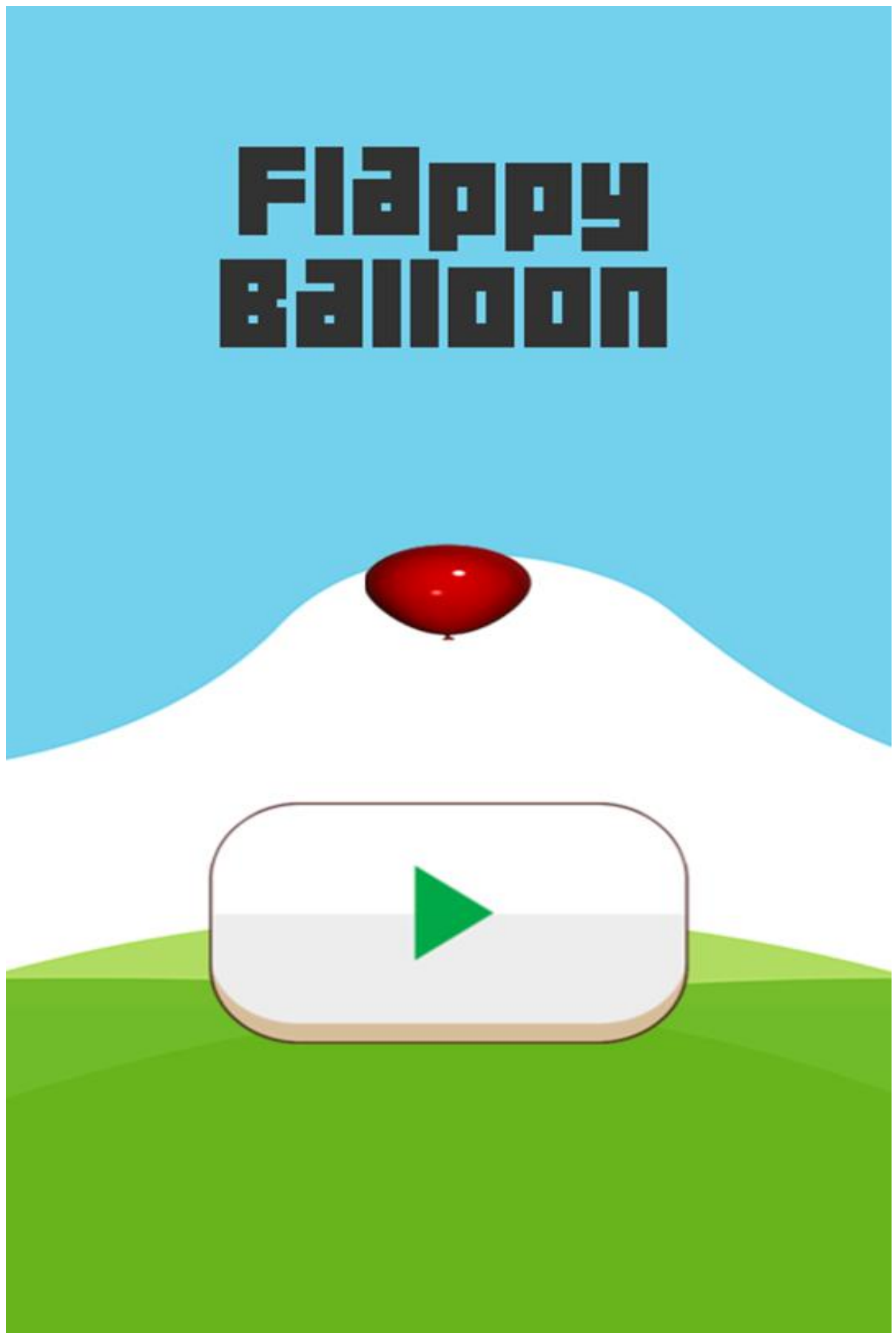
### 4.1 Biên dịch

Biên dịch phần mềm game ra hệ điều hành điện thoại. Như trong bài này, chúng em biên dịch ra hệ điều hành Android.

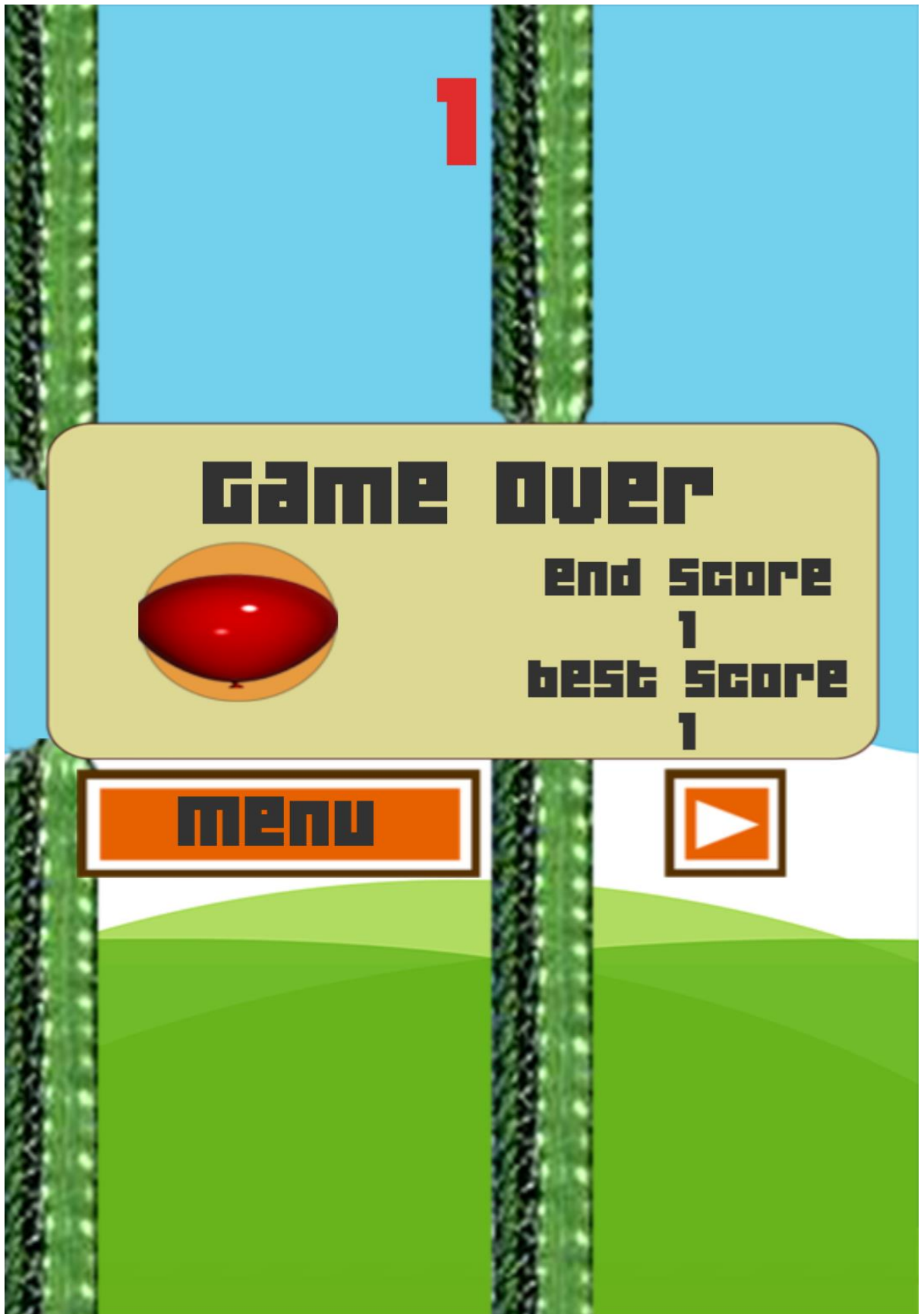


**Hình 4.1. Giao diện biên dịch**

Giao diện ứng dụng sẽ trông như thế này:



**Hình 4.2. Giao diện bắt đầu game**



Hình 4.3. Thông báo kết quả game

## **CHƯƠNG 5: KẾT LUẬN**

### **5.1.Ưu điểm**

Cơ bản nắm được các bước xây dựng một quy trình làm game.

### **5.2.Hạn chế**

Kỹ năng viết mã lệnh còn sơ sài, chưa hiểu sâu

Chưa có chức năng tạo cấp độ cho người chơi

Chưa xây dựng được các setting để người chơi điều chỉnh

Thời gian gấp rút không thể hoàn thành sản phẩm chất lượng.

### **5.3. Các kỹ năng học được**

Nâng cao khả năng làm bài tập nhóm.

Có thể xây dựng được 1 chương trình game cơ bản

Có cái nhìn tổng quan về lập trình game

## **TÀI LIỆU THAM KHẢO**

1. HƯỚNG DẪN GAME FLAPPY BIRD VỚI UNITY 3D TẠI  
KHOAPHAM.VN: [https://www.youtube.com/watch?v=\\_qPRDKmpEKk&list=P  
LzrVYRai0riRwq876NCjZuulv5BjuDCBk](https://www.youtube.com/watch?v=_qPRDKmpEKk&list=P<br/>LzrVYRai0riRwq876NCjZuulv5BjuDCBk) có truy cập lần cuối là 18/01/2018