BOR3

**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 Bird 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 BIRD**

**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*

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 Bird-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.*

**CHƯƠNG 1: KẾ HOẠCH THỰC HIỆN**

**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.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 Bird”**

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

Bảng phân công chi tiết

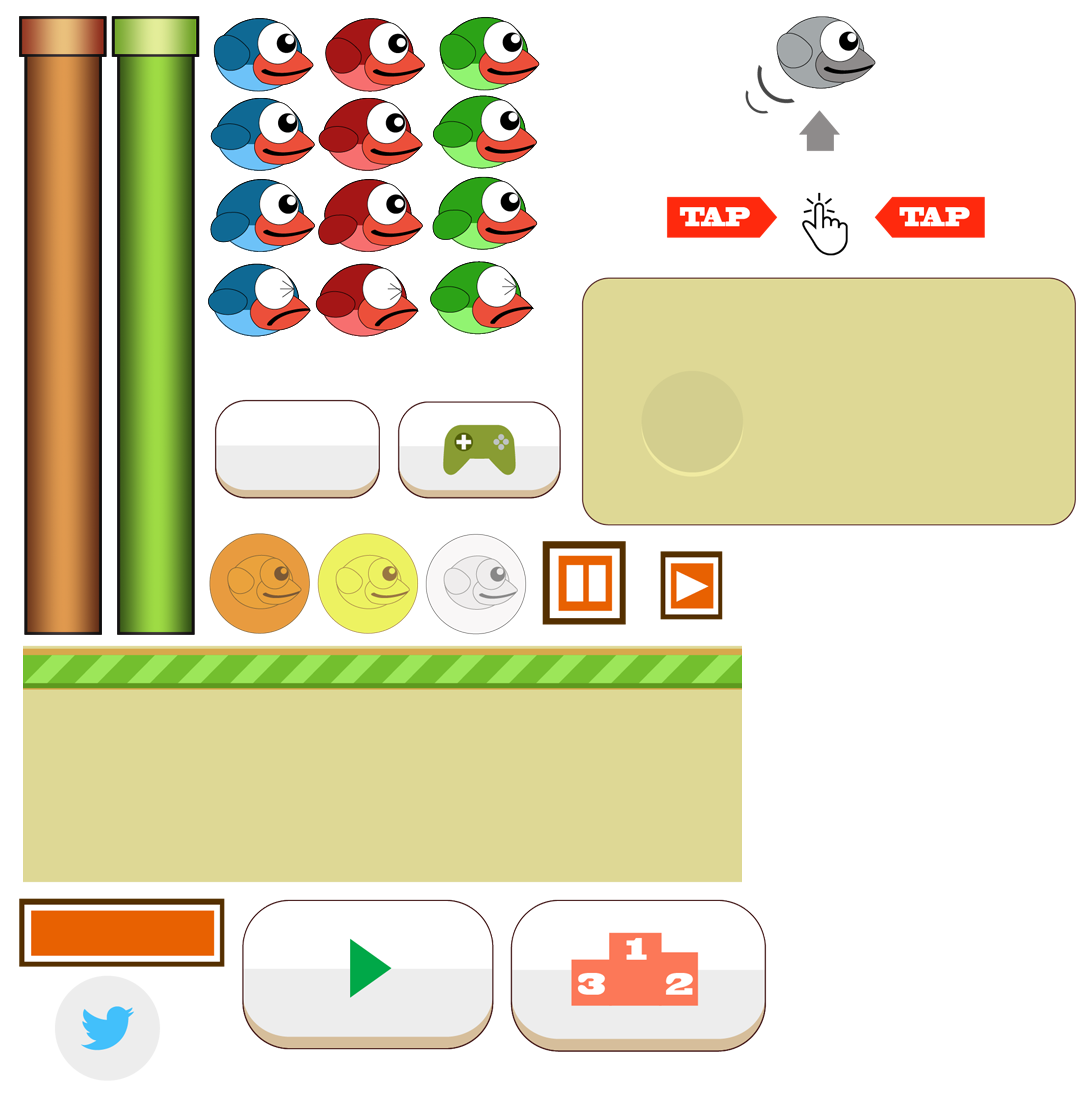
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stt | Công việc | Thời gian thực hiện | Người thực hiện | Ghi chú |
| 01 | Lên ý tưởng,tìm hiểu cách dùng Unity | 25/01/2017-1/1/2018 | Hiếu  Hoàng  Đăng |  |
| 02 | Xây dựng nhân vật,viết code tương tác game | 1/1/2018  -8/1/2018 | Hoàng  Hiếu |  |
| 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 2:XÂY DỰNG GIAO DIỆN,VIẾT MÃ LỆNH GAME**

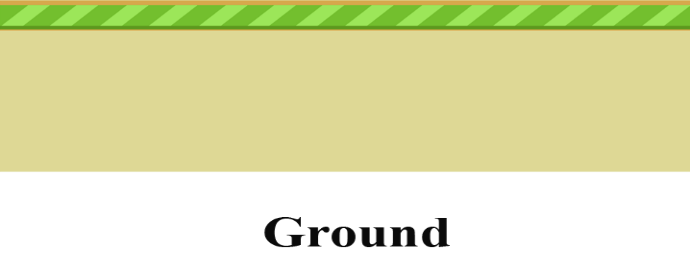
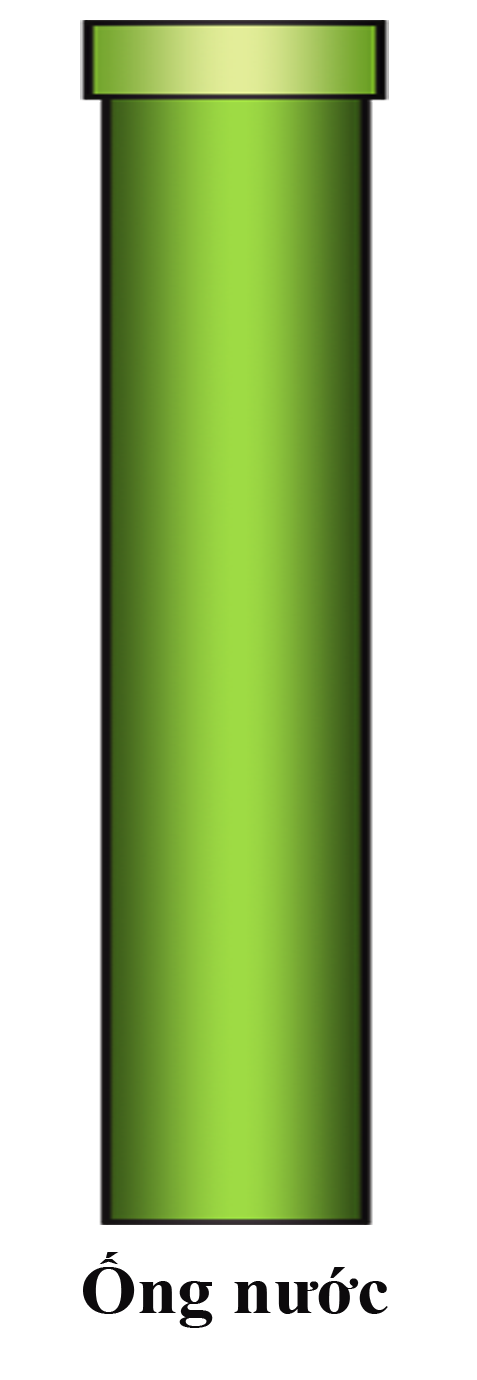
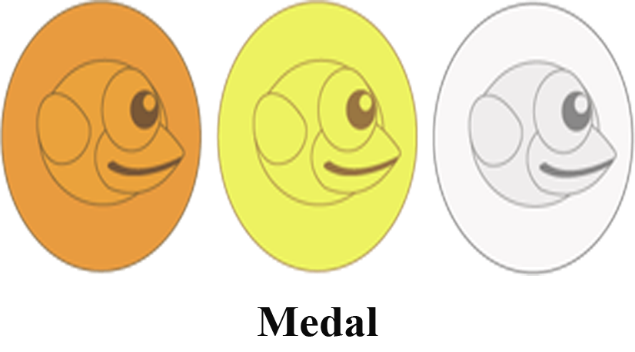
**1.Ý tưởng**

Xây dựng hình tượng chú chim bay đi qua các ống nước.Với mỗi lần bay qua hai ống nước sẽ được cộng một điểm,nếu chạm vào ống nước 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

**2.Xây dựng nhân vật**

****

Hình 1.Tất cả các nhân vật trong game



Hình 2. Nhân vật chính được sử dụng trong game

**3.Viết mã lệnh cho game**

**a.Nhân vật**

**1.Background**



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

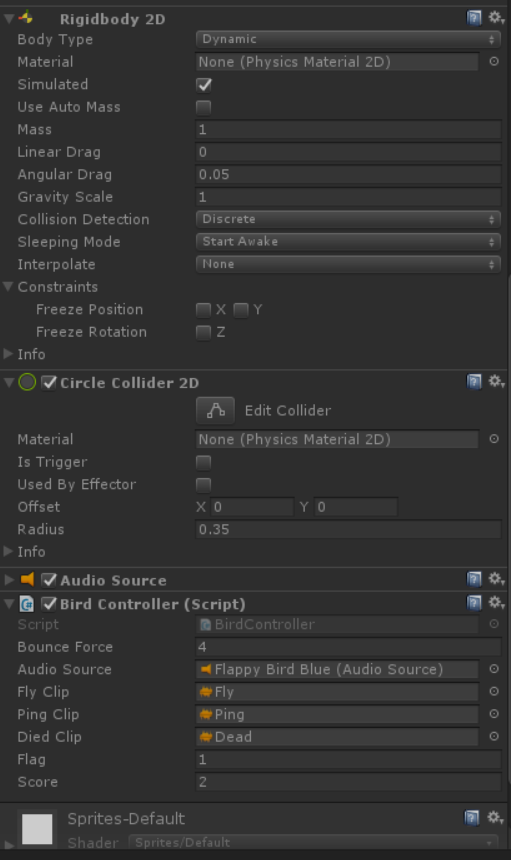
float worldWidth = worldHeight \* Screen.width/Screen.height;//10 \* 203/339

tempScale.y = worldHeight / height;

tempScale.x = worldWidth / width;

transform.localScale = tempScale;

**2.Chú chim vỗ cánh bay:Bird Animation**



Có hai thành phần quan trọng là **Rigidbody 2D** và **Circle Collider 2D** xác định trọng lực của chim và độ va chạm của chim với thành ống

**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 thành ống

**BirdController:**viết mã C# cho chú chim

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 ống 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");

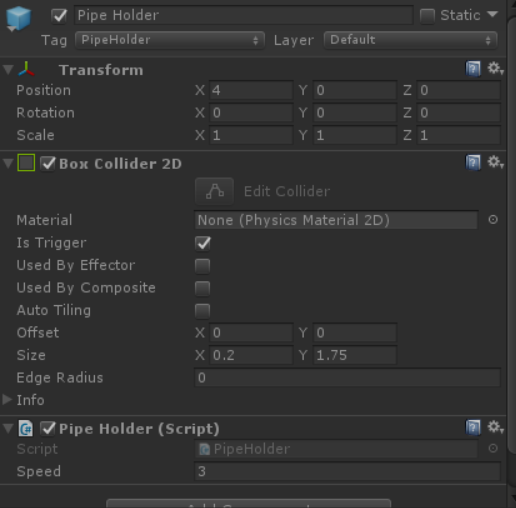
}

if (GamePlayController.instance != null) {

GamePlayController.instance.\_BirdDiedShowPanel (score);

}}}

**3.Ống nước**



Ống nước bao gồm hai ống:ống trên và ống dưới

**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;

}

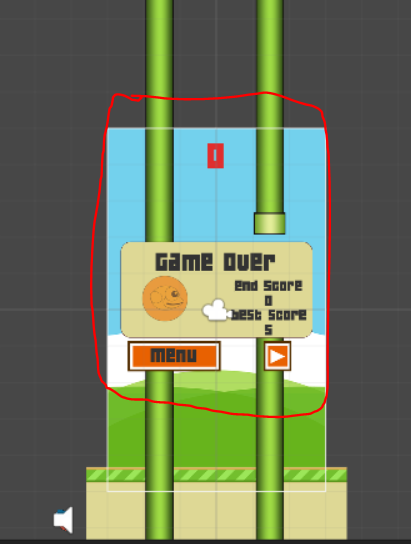
void OnTriggerEnter2D(Collider2D target){

if (target.tag == "Destroy") {

Destroy (gameObject);

}}}

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



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 ());

}

**b.Quản lý game**

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



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);

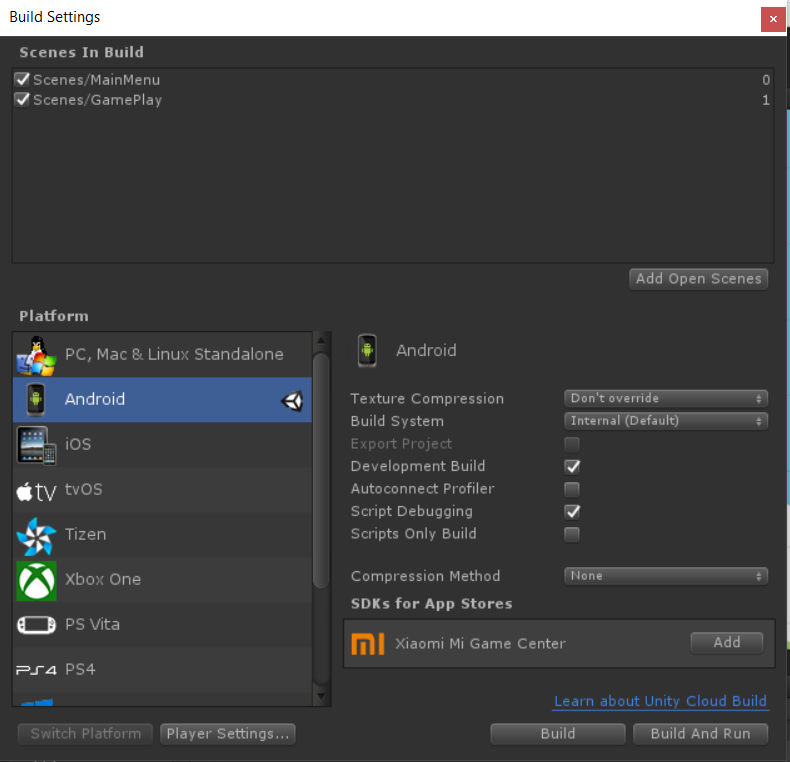
}

bestScoreText.text = "" + GameManager.instance.GetHighScore();

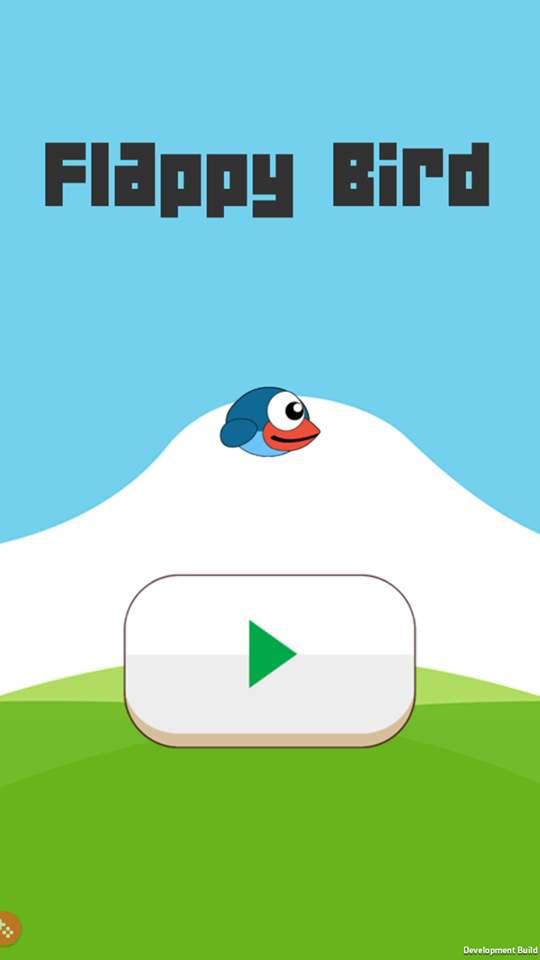
}

**CHƯƠNG 3:BIÊN DỊCH VÀ KIỂM THỬ**

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



Giao diện ứng dụng sẽ trông như thế này:



**CHƯƠNG 4 KẾT LUẬN**

1. **Ưu điểm**

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

1. **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.

1. **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