开发文档报告

注：由于源码文件过大，请在百度网盘内下载

<https://pan.baidu.com/s/1CcWizZDVjsRJ0dwE_gk0qg>

由于视频制作软件问题，导致视频出现拖影，敬请谅解！

1. 整体架构

游戏使用Unity引擎开发，分为level、spring、summer、autumn、winter四个场景。其中GameControl类用于控制游戏中失败等游戏过程。

1. 游戏特色

游戏通过精美的画面、轻柔的配乐，巧妙的设计唤起大家对于童年无忧无虑生活的回忆，仿照“FlappyBird”的玩法，旨在使玩家在快节奏的生活中得到放松。游戏设计了不同难度关卡，不同能力的角色，提供一定深度的游戏体验

1. 关键技术及实现
2. 飞机的飞起

if (Input.GetMouseButtonDown(0)&&!GameControl.isDead)

{

rb2d.velocity = Vector2.zero;

rb2d.AddForce(new Vector2(0, upForce));

}

//防止飞机飞出屏幕

if (transform.position.y >= 4.11f)

{

rb2d.velocity = new Vector2(0f, -1f);

}

1. 障碍物
   1. 障碍物移动，每经过一段时间后加速

if (Time.timeSinceLevelLoad - lastRankTime > rankUpCycle && !GameControl.isDead && lastRankTime <= 180)

{

xspeed \*= 1.3f;//每经过rankupperiod后加速

lastRankTime = Time.timeSinceLevelLoad;

rb.velocity = new Vector2(xspeed, 0);

}

* 1. 普通障碍物生成，每经过一段时间后生成的障碍物数量增多

if (Time.timeSinceLevelLoad - lastGenTime >= genTime && !GameControl.isDead)

{

lastGenTime = Time.timeSinceLevelLoad;

rocks[rockIndex].transform.position = new Vector2(right, Random.Range(ceil, floor));

rocks[rockIndex].SetActive(true);

rockIndex++;

if (rockIndex > num - 1)

rockIndex = 0;

if (Time.timeSinceLevelLoad - lastRankUpTime > 30 && genTime >= 2.5)

{

lastRankUpTime = Time.timeSinceLevelLoad;

genTime -= 1.5f;

}

}

* 1. “上下”型障碍物生成，随机得到生成位置在上方或下方

if (Random.value > 0.5)

rocks[rockIndex].transform.position = new Vector2(right, Random.Range(ceil, floor));

else

rocks[rockIndex].transform.position = new Vector2(right, Random.Range(-floor, -ceil));

}

* 1. 可变大小的障碍物生成

1. if (Time.timeSinceLevelLoad - lastGenTime >= genTime && !GameControl.isDead && (rocks[rockIndex].transform.position.x <= -11 || !rocks[rockIndex].activeSelf))
2. {
3. lastGenTime = Time.timeSinceLevelLoad;
4. float rate = Random.Range(2f, 3f);
5. rocks[rockIndex].transform.position = new Vector2(right, (float)(-4.43+rate\*0.59));
7. rocks[rockIndex].transform.localScale = new Vector2(rate,rate);
8. rocks[rockIndex].SetActive(true);
9. rockIndex++;
10. if (rockIndex > num - 1)
11. rockIndex = 0;
12. if (Time.timeSinceLevelLoad - lastRankUpTime > 15 && genTime >= 3.5f)
13. {
14. lastRankUpTime = Time.timeSinceLevelLoad;
15. genTime -= 0.5f;
16. }
17. }
18. 星星曲线移动，并在一段时间内生成

if (!GameControl.isDead&&!star.activeSelf && Time.timeSinceLevelLoad - activetime > 3f)

{

star.SetActive(true);

star.transform.position = new Vector2(10.5f, 0f);

starttime = Time.timeSinceLevelLoad;

A = Random.Range(3f, 2.5f);

if (Random.Range(0f, 1f) > 0.5)

A = -A;

n = Random.Range(1.2f, 1.7f);

}

if (!GameControl.isDead&&star.activeSelf)

{

rb2d.velocity = new Vector2(-5f, 0);

star.transform.position = new Vector2(star.transform.position.x, A\* Mathf.Sin((Time.timeSinceLevelLoad-starttime)\*n));

if (star.transform.position.x < -11f)

{

star.SetActive(false);

activetime = Time.timeSinceLevelLoad;

}

}