



四川大学
SICHUAN UNIVERSITY



基于Unity与C实现游戏开发的

I Wanna Learn Programming

答辩小组：I Wanna队

小组成员：秦赫璠 谢荣涛 张俊杰 谢俊鑫 崔昊宇

项目地址：<https://github.com/LovePolaBear/I-Wanna-Game.git>



目录 / CONTENTS



01/游戏总体介绍



02/小组成员分工



03/遇到的一些bug



四川大學
SICHUAN UNIVERSITY



游戏总体介绍

地图设计与游戏测试

游戏起源

基本操作、元素

特殊元素

地图展示

测试中发现的BUG

——秦赫璠

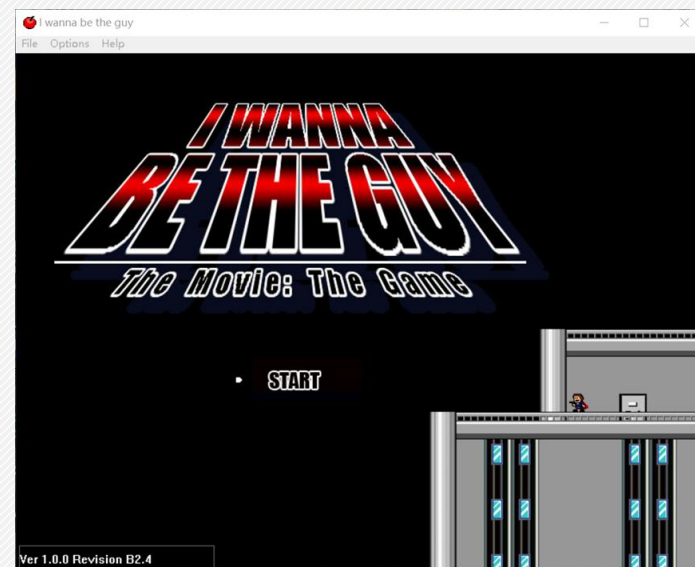


地图设计与游戏测试

游戏起源

《I Wanna Be the Guy》是由Mike "Kayin" O'Reilly制作的并由2007年发布的一部游戏，并有《I wanna be the guy: Gaiden》（外传）及 其他同人 等衍生作品。

现在，I Wanna系列已成为一个2D横版过关游戏系列。该游戏系列以高精度操作与超多陷阱而闻名，经常使玩家死亡数以不寻常的速度增长，因而很多玩家常常因挫败感中途放弃。但由于通关该系列游戏能产生极大的成就感，I Wanna系列仍然存在一个相对稳定的玩家群体并深受他们欢迎。





地图设计与游戏测试

基本元素

Kid: 游戏中可操控主角;

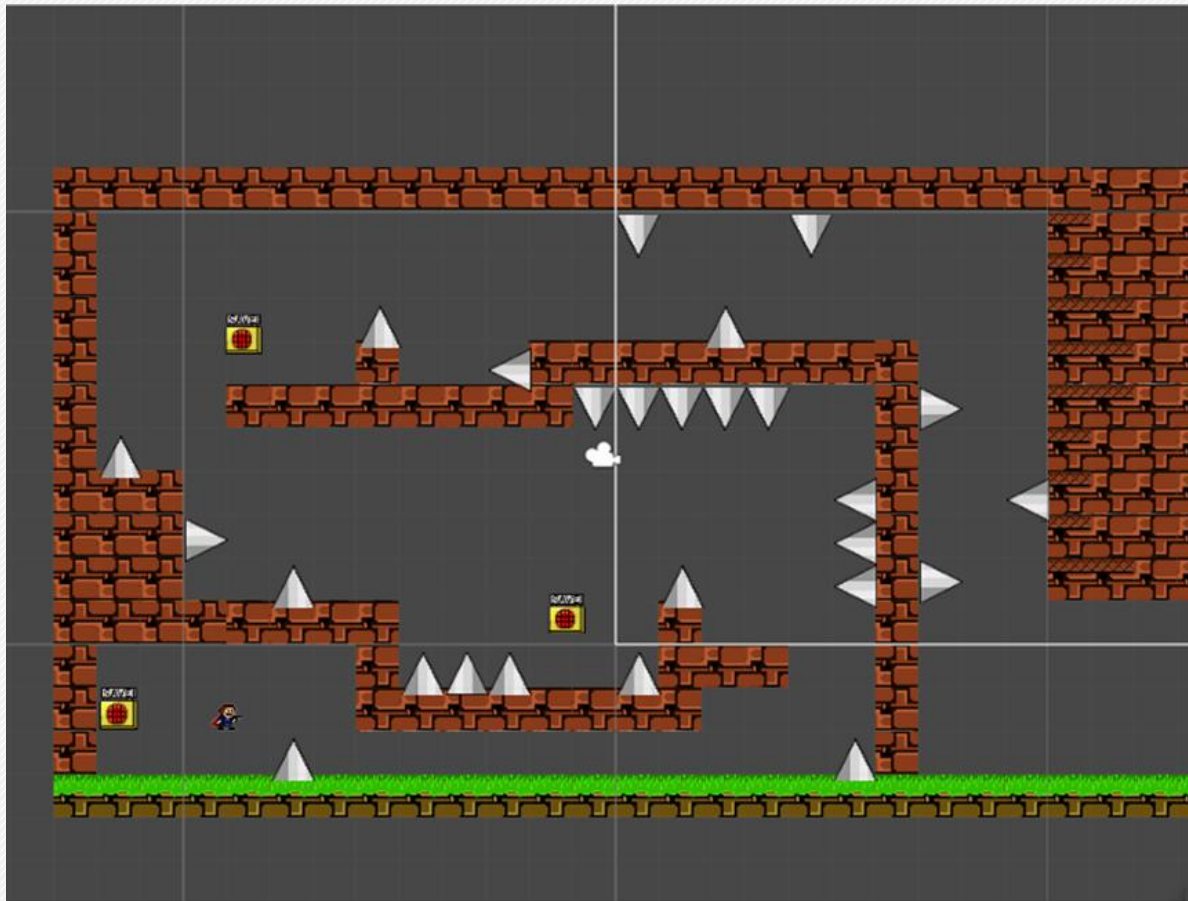
砖块;

刺、苹果: 触碰即死亡;

存档点: 可保存kid当前位置,

不保存机关;

子弹: 可用于存档与攻击boss



基本操作

←→键: 左右移动;

R键: 重来;

Z键: 射击;

Shift: 跳跃 (可二段跳)



地图设计与游戏测试

特殊元素

会飞的刺、苹果：kid在碰到触发器之后刺或苹果会向kid所在位置飞过去，而我们需要先触发陷阱后及时闪避，然后才能继续前进。

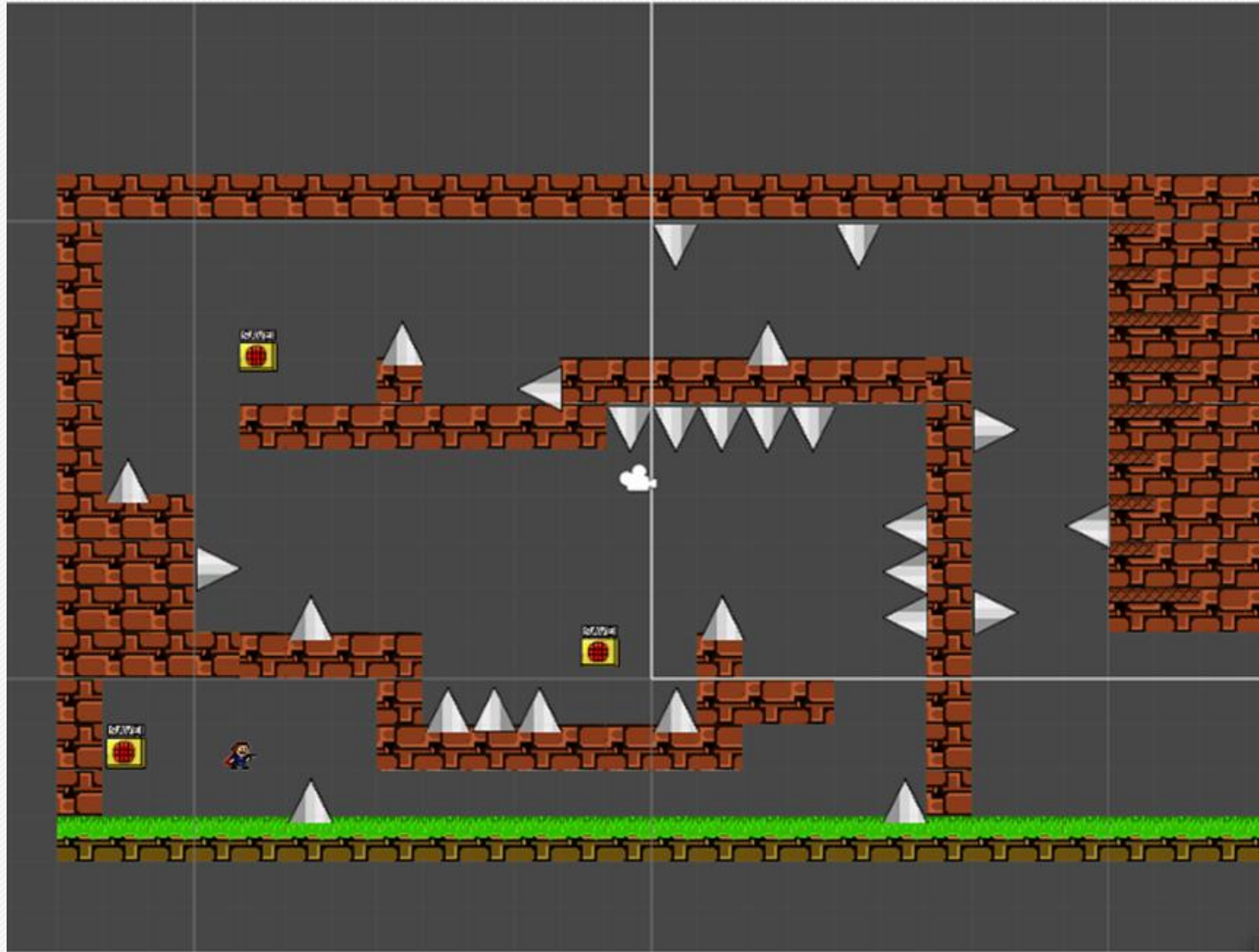
陷阱存档：与正常存档有细微差别若不认真观察就会中招。在射击后会将kid直接杀死或者困住，然后只能重来。



地图设计与游戏测试

地图展示

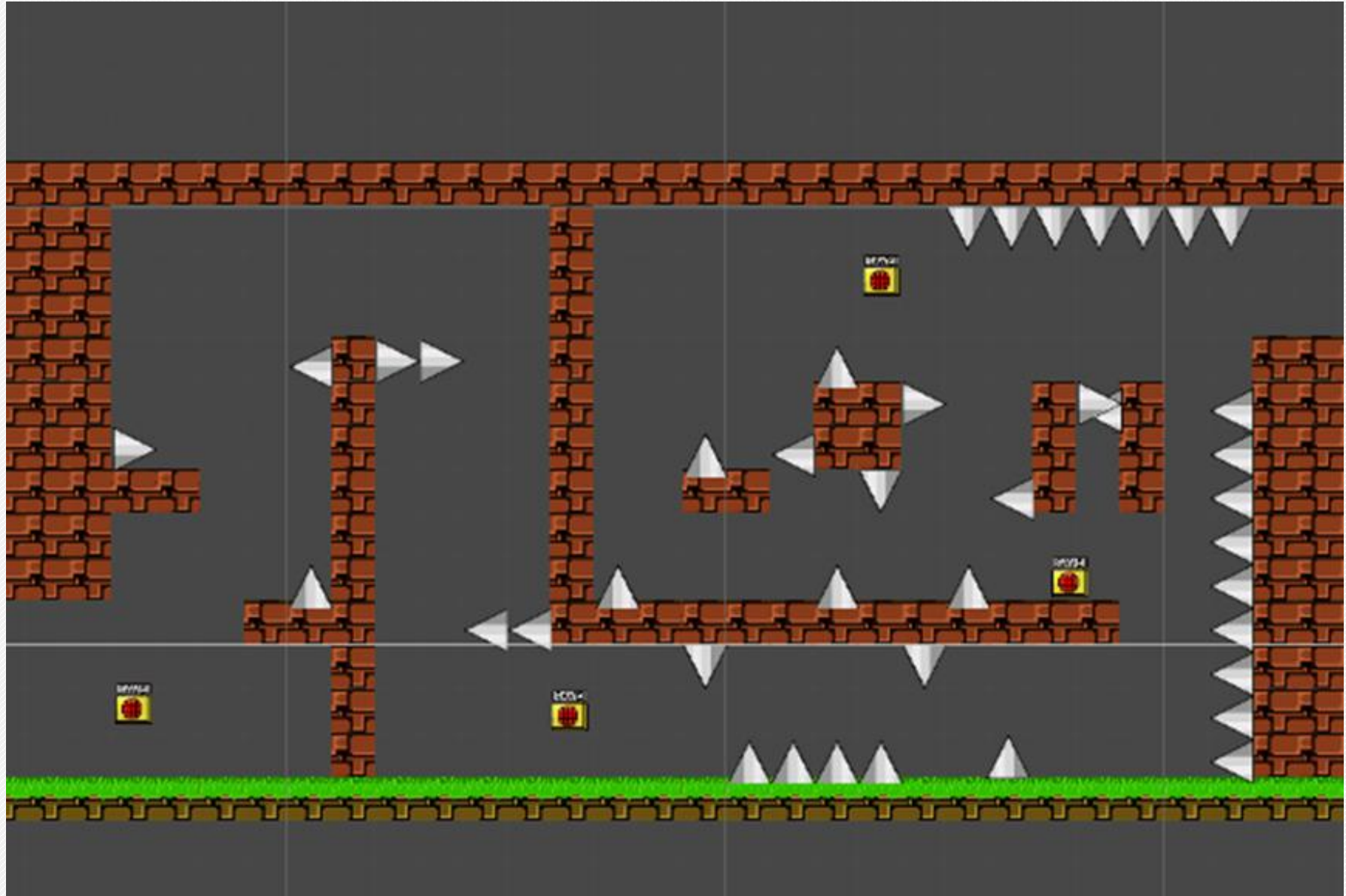
目的：让用户感受到困难，但能过关





地图设计与游戏测试

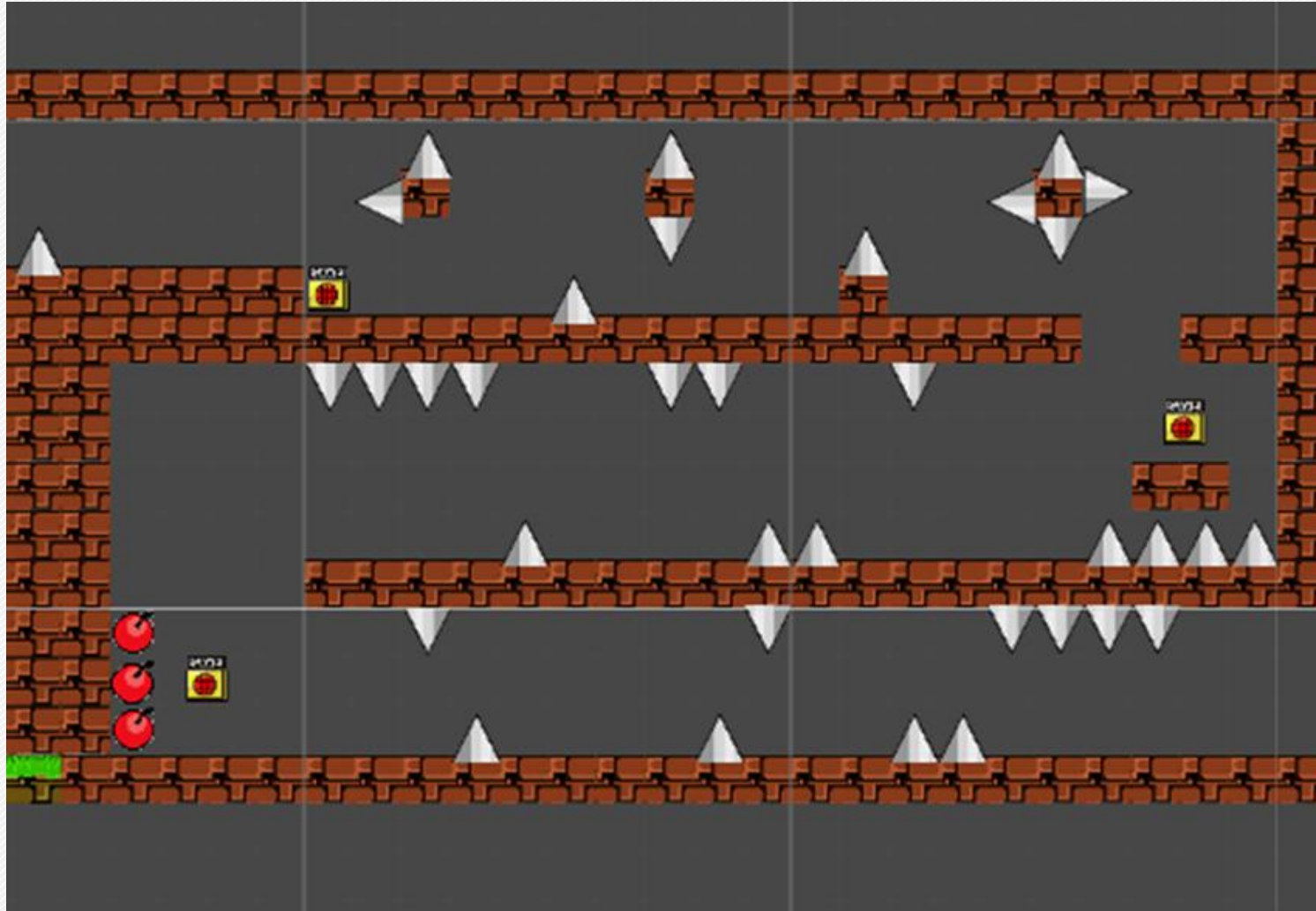
地图展示





地图设计与游戏测试

地图展示



地图设计来源:

I wanna be the GGM

I wanna enjoy the game



那就来...演示一下吧。



测试中发现的BUG

1. 摄像头没有切换位置，导致人物过去了镜头没过去
2. 人物穿墙问题
3. 刺的判定形状是正方形

后面将讲述对其的修正。



四川大學
SICHUAN UNIVERSITY



文档编写

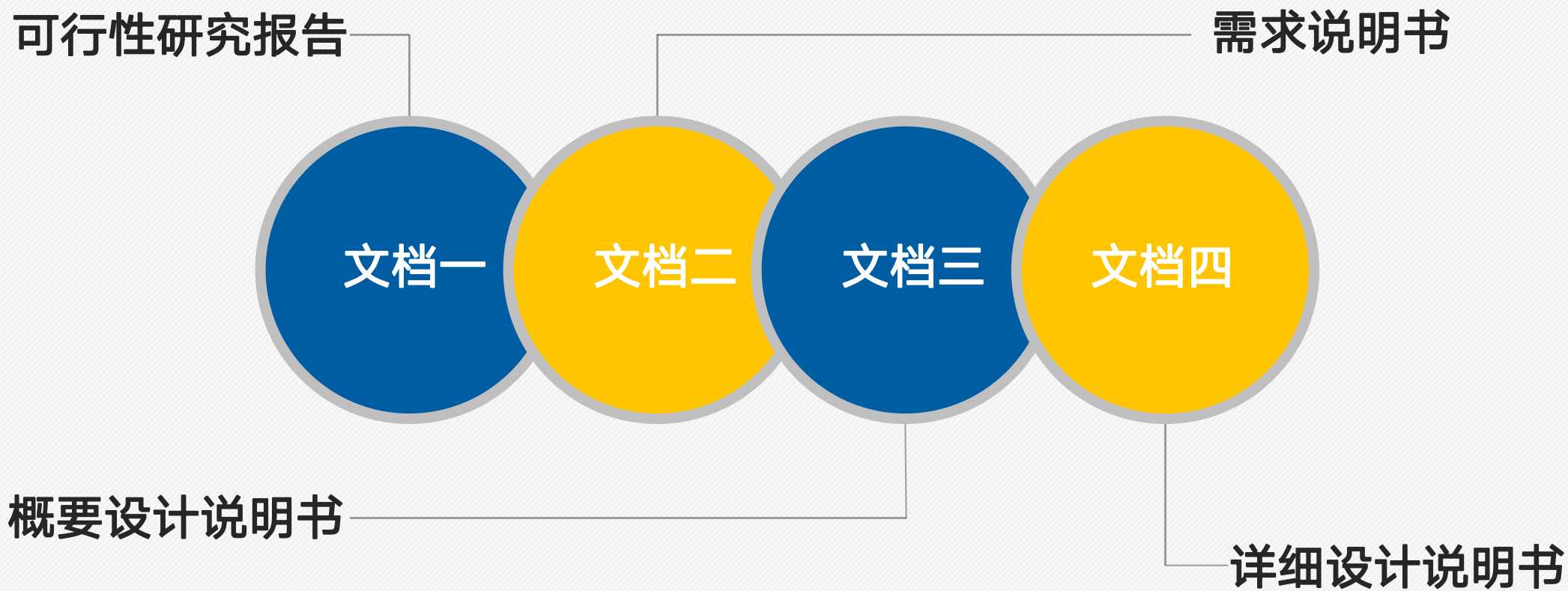
工作内容

编写历程

——谢俊鑫



文档编写





- 可行性研究报告

产生做这样一个游戏的灵感来源：每个同学在小的时候都接触过超级马里奥这一类的闯关冒险游戏，这样的游戏让玩家在不断的越过陷阱通过关卡中取得成就感。放松身心。在上了大学接触计算机专业后发现做出这样的游戏似乎是我们能够实现的一个任务，就正好趁着这个暑假编程实战的机会在小组商讨下决定做一个I wanna系列的游戏。！



- 需求说明书的编写

在确定好要做什么游戏了之后，我们需要分析在这个游戏当中要实现什么功能以及这个游戏要求我们要做到些什么。所以在大家的商讨之下我们大致列出了一个总的需求。

键盘输入：←	游戏中角色动作：向左移动
键盘输入：→	游戏中角色动作：向右移动
键盘输入：shift	游戏中角色动作：向上跳跃
键盘输入：R	游戏中角色动作：存档复活
键盘输入：Q	游戏中角色动作：原地自杀
键盘输入：Z	游戏中角色动作：射出子弹
游戏中角色进入传送门	地图变换
游戏中触碰尖刺	角色原地死亡



- **概要设计说明和详细设计说明**

在明确了方向以及我们游戏的需求过后就要开始思考具体的设计思路和逻辑。我们开始编写概要设计说明书和详细设计说明书，在这两份文档中包含了总体设计、运行逻辑、运行要求等内容，并在详细设计说明书完成后交由小组内的开发人员观看，并按此说明书进行游戏开发，最终完成整个流程。！



四川大學
SICHUAN UNIVERSITY



游戏框架的构建以及 部分功能的实现与 Debug

构建游戏框架

物品类的实现

后期bug的修复

辅助事务

——张俊杰



分工部分——张俊杰

游戏框架的搭建

整体构架程序逻辑，规定相关参数数据类型，确定游戏大体框架。

物品类的实现

编写物品类的脚本逻辑，直接用于刚体绑定部分



后期bug的修复

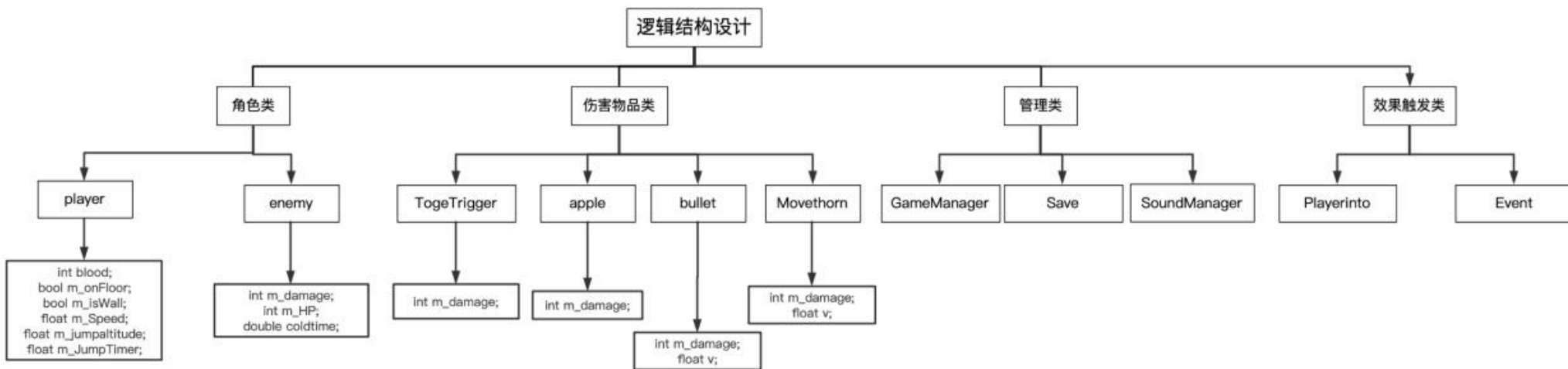
根据秦赫璠提供的bug，与谢荣涛一同修复相应bug

辅助事务

空余时做些杂事，如：游戏素材抠图处理，写一小部分文档



分工部分——张俊杰——游戏框架



player	int blood	外部接口访问
	bool m_onFloor	全局变量访问
	bool m_isWall	全局变量访问
	float m_speed	外部接口访问
	float m_jumpaltitude	外部接口访问
	float m_JumpTimer	外部接口访问
enemy	int m_damage	外部接口访问
	int m_HP	外部接口访问
	double coldtime	外部接口访问

togetrigger	int m_damage	外部接口访问
apple	int m_damage	外部接口访问
bullet	int m_damage	外部接口访问
	float v	外部接口访问
movethorn	int m_dama	外部接口访问
	float v	外部接口访问



分工部分——张俊杰——物品类的实现

```
1 reference
7 | public int m_damage = 10;
0 references
8 | void OnTriggerEnter2D(Collider2D other)
9 | {
10 |     if (other.gameObject.tag == "Player")
11 |     {
12 |         other.SendMessage("BeDamaged", m_damage, SendMessageOptions.DontRequireReceiver);
13 |     }
14 | }
```

静态伤害物体

```
1 reference
7 | public Rigidbody2D m_toge;
1 reference
8 | public float speed = 10f;
0 references
9 | private void OnTriggerEnter2D(Collider2D other)
10 | {
11 |     if(other.tag=="Player")
12 |     {
13 |         Vector2 v = new Vector2(0, speed);
14 |         m_toge.velocity = v;
15 |         Destroy(gameObject);
16 |     }
17 | }
```

加些速度呢...?
飞行伤害物体
完成子弹，飞刺功能!



分工部分——张俊杰——物品类的实现

```
void Beshot(int k)
{
    Vector3 size = new Vector3(3, 3, 1);
    transform.localScale = size;
}
```

子弹射击过后触发些条件...?
陷阱出口完成!

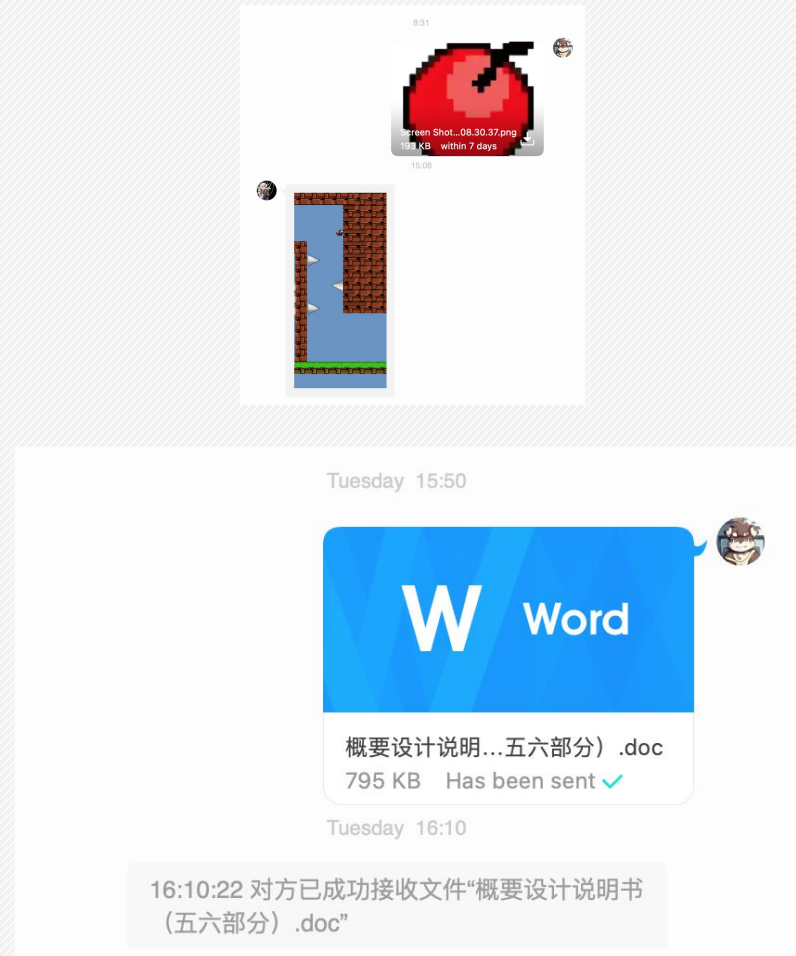
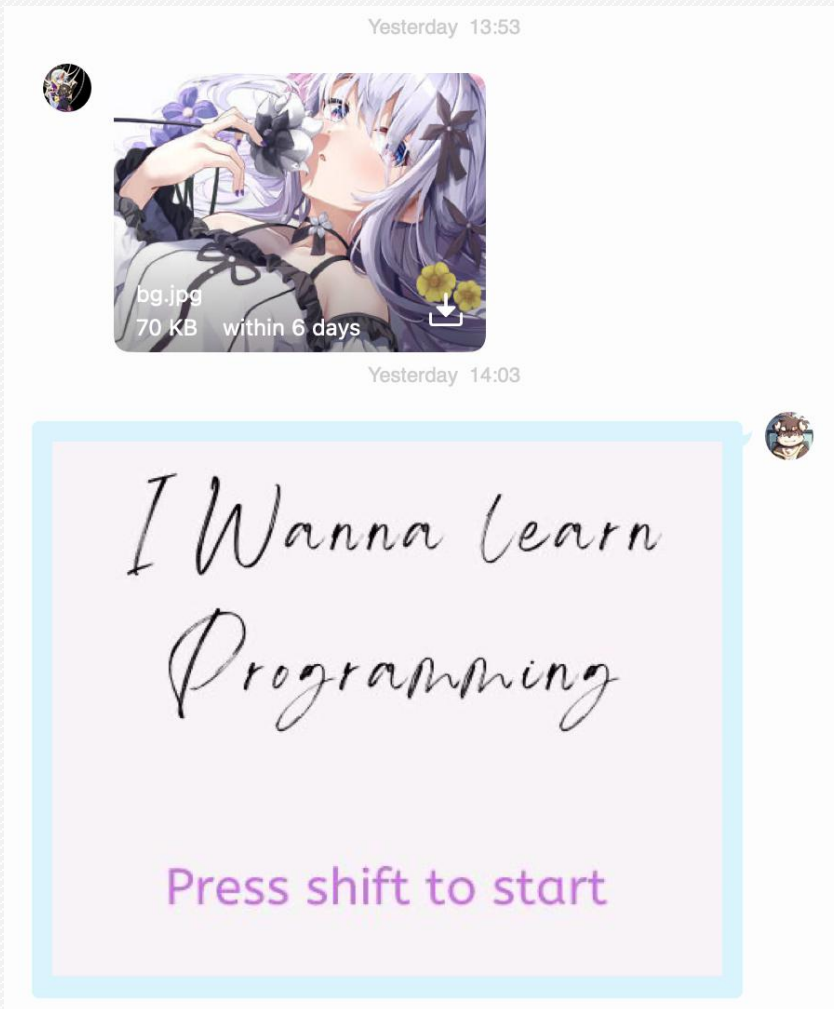
```
7      private void OnTriggerEnter2D(Collider2D other)
8      {
9          if (other.tag == "apple")
10         {
11             Destroy(gameObject);
12         }
13     }
```

那到终点追击的苹果继续
追击咋办...?
结合飞行伤害物品
创建一个新的调用函数触
发条件时销毁苹果



分工部分——张俊杰——辅助事务

人物类实现和地图设计时闲到扣扣图，写一点点文档...





四川大學
SICHUAN UNIVERSITY



人物类实现 修改BUG

Player相关功能

跳跃、射击

玩家死亡

BUG的解决

——谢荣涛



人物类实现&修改BUG

一、左右移动

1.转向

```
private void Flip()
{
    m_FacingRight = !m_FacingRight;
    Vector3 scale = transform.localScale;
    scale.x *= -1;
    transform.localScale = scale;
}
//player朝向
```

调用位置在水平移动



人物类实现&修改BUG

一、左右移动

左右的水平移动经历了两个开发过程

(1) 没有跳跃时期

直接定义一个变量V使得其速度等于V即可 (V+ -决定方向)

(2) 有跳跃时期

需要定义一个向量vector2 (v1,v2) 其中v1是速度, v2为竖直方向上速度

2.移动

```
Vector2 v = m_ve.velocity;  
v.x = h * m_Speed * Time.deltaTime; // Time.deltaTime 一帧的时间  
m_ve.velocity = v;
```

判断是否在地面, 判断是否转向以及是否在地面以使得动画正常



一、左右移动

2.移动

```
if (m_FacingRight)
{
    if (h < 0)
    {
        Flip();
    }
    else if (m_isWalled)
    {
        m_anim.SetBool("run", false);
        return;
    }
}
else
{
    if (h > 0)
    {
        Flip();
    }
    else if (m_isWalled)
    {
        m_anim.SetBool("run", false);
        return;
    }
}
```

```
m_anim.SetBool("run", !(h == 0));
```

如果速度不为0，播放动画“run”



人物类实现&修改BUG

二、跳跃

```
if (m_isGrounded)
{
    m_jumpTimes = 1;

    m_isJumping = true;
    m_JumpTimer = 0f;
    m_isGrounded = false;
    m_vec.x = m_ve.velocity.x;
    m_ve.velocity = m_vec;

    //音效
    GameObject.Find("SoundManager").SendMessage("FJumpAudio");
}
//一段跳
else if (m_jumpTimes == 1)
{
    m_jumpTimes = 2;

    m_isJumping = true;
    m_JumpTimer = 0f;
    m_isGrounded = false;
    m_vec.x = m_ve.velocity.x;
    m_ve.velocity = m_vec;

    //音效
    GameObject.Find("SoundManager").SendMessage("SJumpAudio");
}
//二段跳
```



二、跳跃

```
if (m_ve.velocity.y <= -12f)
{
    Vector2 v = m_ve.velocity;
    v.y = -12f;
    m_ve.velocity = v;
}
//最大下落速度
```

同时因为unity常见的穿墙bug，通过限定最大下落速度来保证player不会穿墙而过掉出这个世界



人物类实现&修改BUG

二、射击

在按下“射击”时生成子弹，并在玩家位置生成一个有玩家面向方向速度的子弹。

```
if (Input.GetButtonDown("射击"))
{
    GameObject obj = Instantiate(pfb_bullet, transform.position, Quaternion.identity);
    obj.GetComponent<Rigidbody2D>().velocity = m_FacingRight ? bulletSpeed : -1 * bulletSpeed;
    //音效
    GameObject.Find("SoundManager").SendMessage("SHootAudio");
}
//射击
```

子弹的设计：伤害为10，碰到物体时调用物体的“Beshot”，不要求反馈，然后销毁子弹



人物类实现&修改BUG

三、玩家死亡

```
public int m_HP = 10;
public GameObject ui_GameOverImage;
void BeDamaged(int damage)
{
    m_HP -= damage;
    if (m_HP <= 0)
    {
        // 玩家死亡
        Destroy(gameObject);
        ui_GameOverImage.SetActive(true);
        //音效
        GameObject.Find("SoundManager").SendMessage("pauseLevelAudio");
        GameObject.Find("SoundManager").SendMessage("DEathAudio");
    }
}
```

当玩家碰到有伤害的物体时，调用该函数，如果 $m_HP < 10$ ，销毁玩家的图像，并将`Ui_GameOverImage` 设为存在。达到玩家死亡场景的实现。



人物类实现&修改BUG

四、BUG的解决（以墙和地面的区分为例）

```
private bool IsGrounded()
{
    Debug.DrawRay(transform.position, Vector2.down, Color.green);
    RaycastHit2D hit = Physics2D.Raycast(transform.position, Vector2.down, m_groundCheckDistance, m_groundLayer);
    if (hit.collider != null)
    {
        return true;
    }
    return false;
}
//地面判断

private bool IsWalled(float dir)
{
    RaycastHit2D hit1 = Physics2D.Raycast(m_headCheck.position, dir * Vector2.right, m_wallCheckDistance, m_groundLayer);
    RaycastHit2D hit2 = Physics2D.Raycast(m_footCheck.position, dir * Vector2.right, m_wallCheckDistance, m_groundLayer);
    if ((hit1.collider == null) && (hit2.collider == null))
    {
        return false;
    }
    return true;
}
//墙面判断
```



四川大學
SICHUAN UNIVERSITY



技术测试

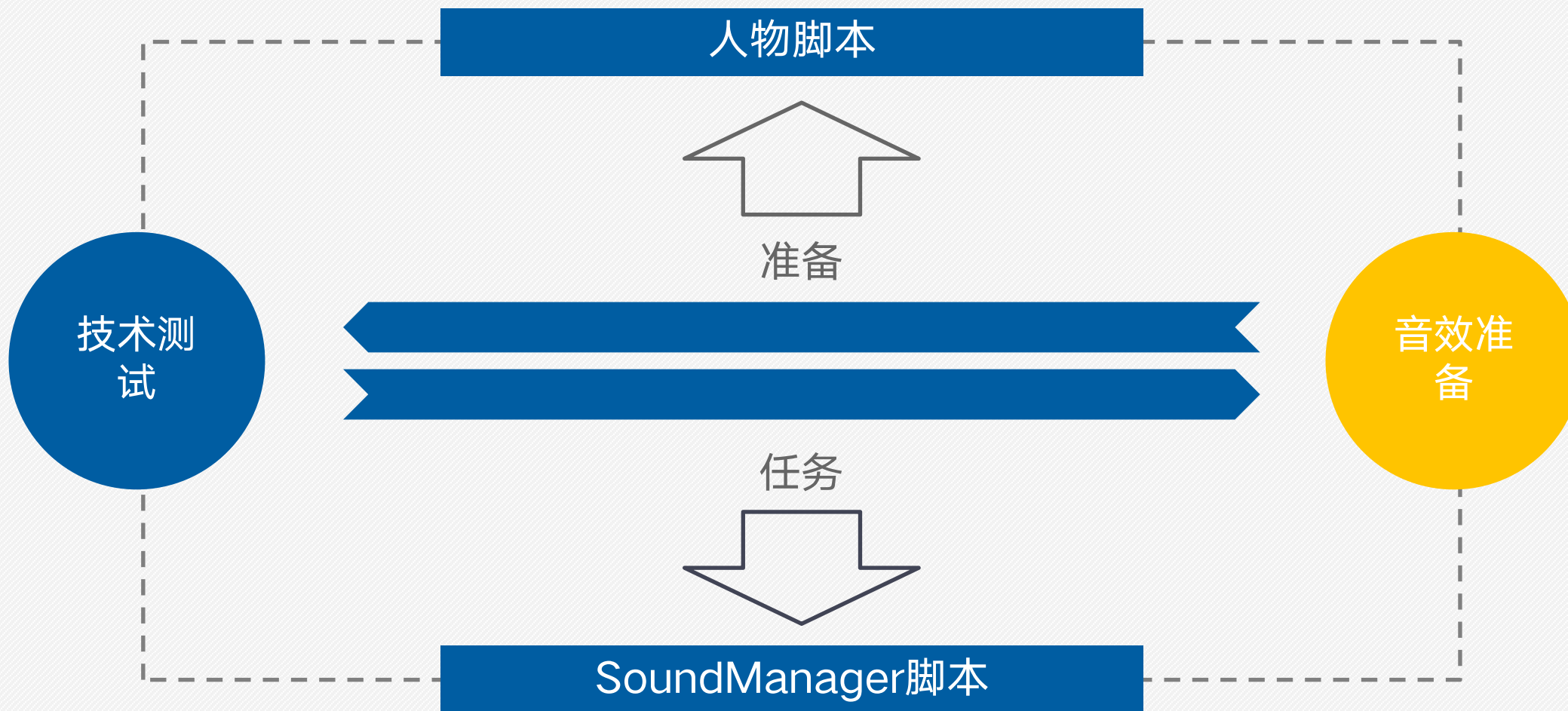
音效制作

技术测试与音效制作

——崔昊宇



技术测试和音效准备

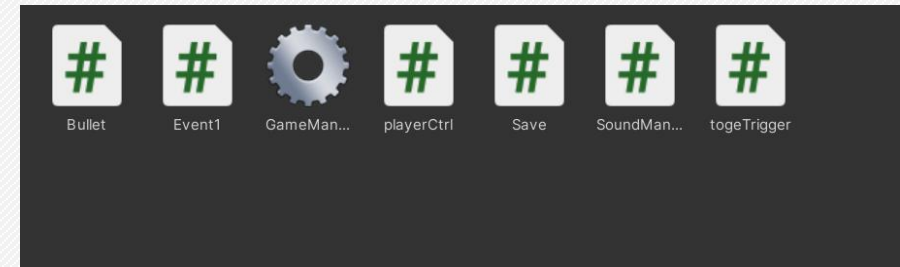
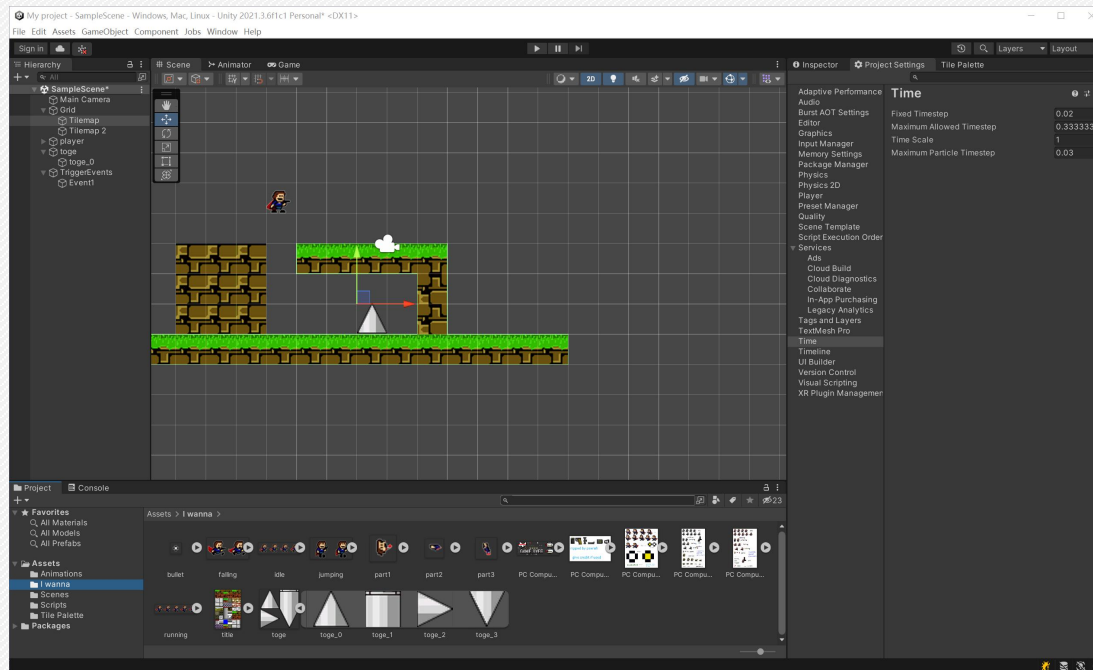




技术测试和音效准备

技术测试

在学习了Unity的一些基本操作后，对人物的控制脚本进行了编写测试，理解了控制逻辑





技术测试和音效准备

音效准备

```
public static SoundManager instance;  
public AudioSource AudioSource; //音效  
public AudioSource backaudio; //背景音乐  
[SerializeField]  
private AudioClip fjumpAudio, sjumpAudio, deathAudio, backAudio, bossAudio, shootAudio; //声明，在unity中进行赋值
```




技术测试和音效准备

音效准备

```
private void Awake()  
{  
    if (instance != null && instance != this)  
    {  
        Destroy(this.gameObject); //避免多个不可删除的该组件出现，也就是单个个例  
        return;  
    }  
    else  
    {  
        instance = this;  
    }  
    DontDestroyOnLoad(gameObject); //不允许删除，因为我需要保持音乐进度，切换场景，重新开始游戏都需要保存音乐  
    backaudio = gameObject.AddComponent<AudioSource>(); //获取组件  
    AudioSource = gameObject.AddComponent<AudioSource>();  
    backaudio.clip = backAudio;  
  
    instance.AudioSource.volume = 0.99f; //音量大小  
    instance.backaudio.volume = 0.1f;  
    instance.backaudio.loop = true; //背景音乐循环播放  
    StartLevelAudio(); //开始时自动播放音乐
```



技术测试和音效准备

音效准备

```
public void fJumpAudio()//第一次跳跃音效
{
    AudioSource.clip = fjumpAudio;
    AudioSource.Play();
}
```

```
if (m_isGrounded)
{
    m_jumpTimes = 1;

    m_isJumping = true;
    m_JumpTimer = 0f;
    m_isGrounded = false;
    m_vec.x = m_body.velocity.x;
    m_body.velocity = m_vec;

    GameObject.Find("SoundManager").SendMessage("fJumpAudio");
}
```

跨脚本调用fJumpAudio函数



四川大學
SICHUAN UNIVERSITY



遇到的一些Bug

bug1

bug2

其余bug

——张俊杰，谢荣涛



遇到的一些Bug——bug1

1 reference

```
public GameObject m_player;
```

3 references

```
public static float count = 0;
```

3 references

```
public GameObject ui_GameStartImage;
```

```
// Start is called before the first frame update
```

```
if (Input.GetKeyDown(KeyCode.RightShift))
```

```
{
```

```
    count++;
```

```
    ui_GameStartImage.SetActive(false);
```

```
}
```

尝试引入一个count全局变量，增加一个判断条件

但仍然存在一直出现ui的现象

尝试每次开始将ui图设成false

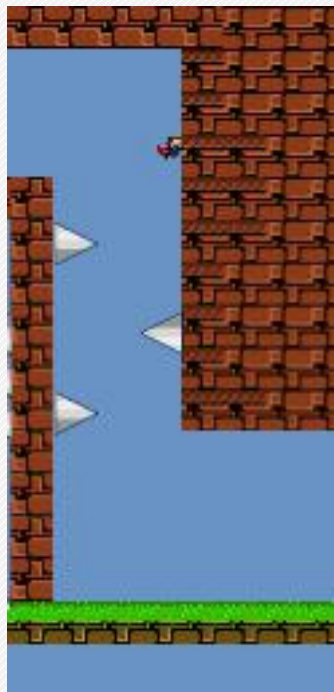
发现ui图会在每次按下r后出现刚开始的界面



遇到的一些Bug——bug2

```
if (m_ve.velocity.y <= -12f)
{
    Vector2 v = m_ve.velocity;
    v.y = -12f;
    m_ve.velocity = v;
}
```

增加限制人物速度函数，问题解决（注：下落时速度为负数）



无用，考虑是否是因为人物速度过快造成

尝试提高帧率，使地图刷新频率加快

人物从高处落下会卡到地下



遇到的一些Bug——其他bug

过程中其实还有许许多多小问题...

- 1.刚体触发器的名称首字母大写规则，脚本名称不匹配
 - 2.摄像机不改变坐标：重新设置同类型向量，通过向量更改
 - 3.地刺触发器形状不与贴图形状贴合
 - 4.音效触发但不发生
 - 5.墙地二象性
 - 6.首次打开存档不为首地址
 - 7.场景转换触发器记录当前位置导致无限触发，不能继续操作
-

痛并快乐地改bug！！！



四川大学
SICHUAN UNIVERSITY



感谢您的收看

答辩小组：I Wanna队

小组成员：秦赫璠 谢荣涛 张俊杰 谢俊鑫 崔昊宇