

# 魔剑传说

——Tales of evil sword

## 游戏设计报告



小组成员：庞博 杨明亮 董泰佑 张文佳 胡义伟

# 目录

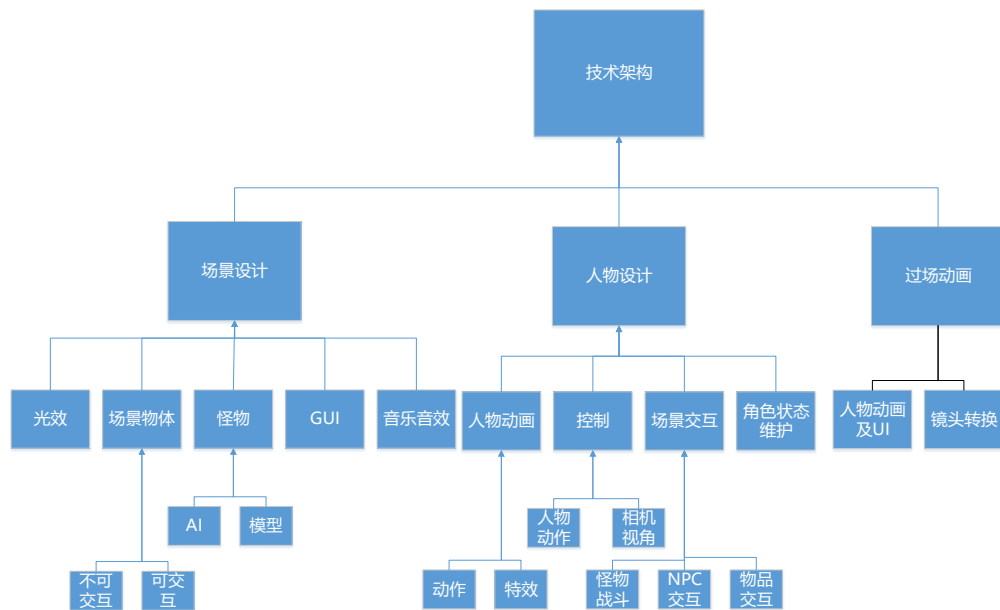
一、整体介绍 .....	4
二、游戏设计架构 .....	5
三、人物系统设计 .....	6
3.1 人物动画.....	6
3.2 控制 .....	7
3.3 场景交互.....	8
3.4 角色状态维护.....	10
四、物品装备系统设计 .....	11
4.1 物品系统设计 .....	11
4.2 技能系统与几点储存.....	14
4.3 技术细节的实现.....	15
五、技能特效设计 .....	19
5.1 魔法效果设计： .....	19
5.2 刀光效果设计 .....	22
六、场景设计及搭建.....	24
6.1 场景光照的设定： .....	25
6.2 碰撞检测的设定.....	26
6.3 场景小地图的添加和绘制 .....	27
七、怪物 AI.....	32
八、过场动画 .....	34
九、游戏剧情 .....	36

十、组员及分工.....	40
十一、附录.....	41
游戏对话剧情： .....	41

# 一、整体介绍

本作是一款西式魔幻风格的动作角色扮演游戏(ARPG)，主要以传说系列、伊苏系列等经典 RPG 游戏的感觉和理念而创作。游戏故事主要讲述的是一名魔剑士少女提拉米苏在偶然得知自己的身世后，为了寻找杀害双亲的凶手，在泰雷诺亚大陆上不断冒险、不断成长的故事。游戏采用 Unity3D 引擎制作，以有趣的人物剧情和爽快的动作打击感为主要卖点，并包含了丰富的物品、技能系统，酷炫的场景，能带给玩家丰富的游戏体验。

## 二、游戏设计架构

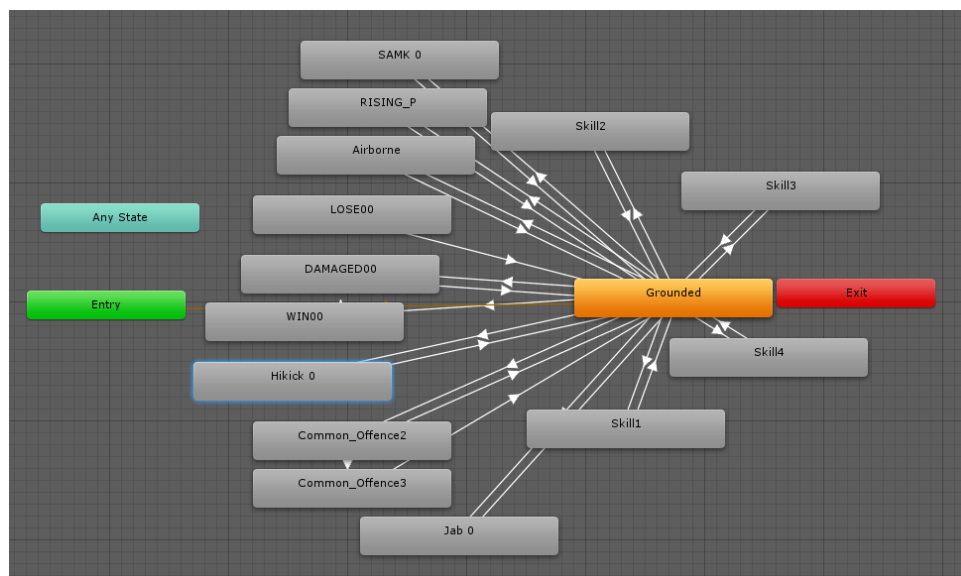


## 三、人物系统设计

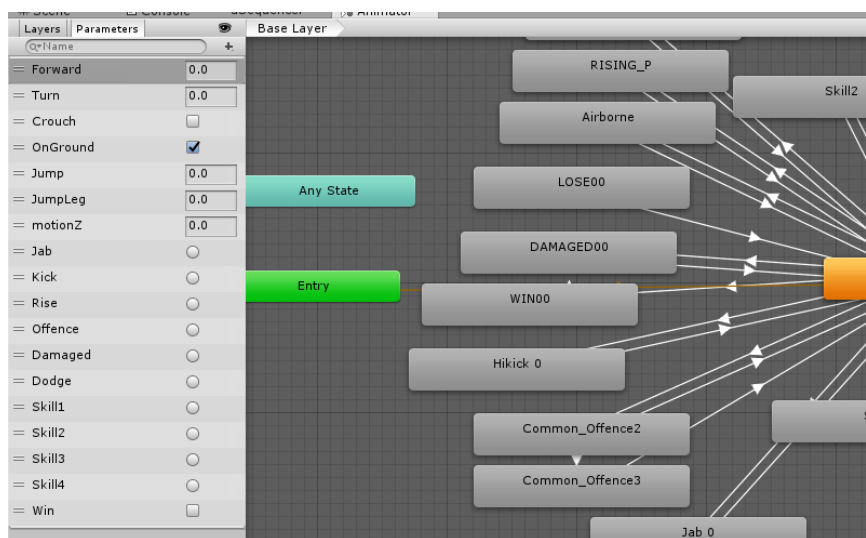
### 3.1 人物动画

Unity 从 4.0 开始引入了一套全新的动画系统 Mecanim，提供了动画重定向的功能以及可视化的 Animator 和 Animation 编辑器，可以很方便地实现把动画从一个角色模型应用到其他模型身上。我们使用 Mecanim 这套系统创建了一些人物动画，同时也将一些创建好的动画片段套用到我们的人物身上，并对每种类型的人物/怪物分别编写了动画流控制脚本，定义好动画的状态切换方向、条件。

我们整合了三个角色的骨骼动画，套用到我们的主角上。如下图所示是我们为主角所设计的动画状态机，一共设计了 4 种攻击动画，4 种技能动画，以及受伤、死亡等状态动画。每个方块代表一个状态，方块之间的有向线段表示状态的迁移，且每条有向线段上都隐含了状态转移的条件和转移时间。



所有状态转移涉及的条件如下图所示，这些条件的值能够动态变化，以此来控制状态的变化：



通过该动画系统，我们就能够很方便地在程序中控制动画状态的流动，以人物受到攻击

播放动画为例，只要使用：animator 类提供的 set 系列函数，即可修改人物的动画状态：

```
public void UpdateAnimator(string animClip ) {

    if ( animClip != null ) {
        m_Animator.SetBool(animClip, true);
    }

}

public void BeAttacked( ) {
    this.UpdateAnimator("Damaged");

    int damage = Random.Range(200, 300);
    bool isCritical = damage > 250;

    attribute.TakeDamage(damage.ToString(), isCritical);

}
```

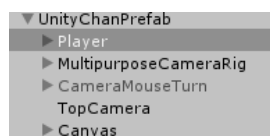
## 3.2 控制

人物控制的实现需要从两个方向考虑：动作的控制和玩家视角的控制。动作的控制主要用来与人物动画进行配合，通过解析用户的按键输入来切换不同的人物动画状态。

相对更复杂一些的是玩家视角，也就是镜头的控制。在此项目中我们打算采用常见的第

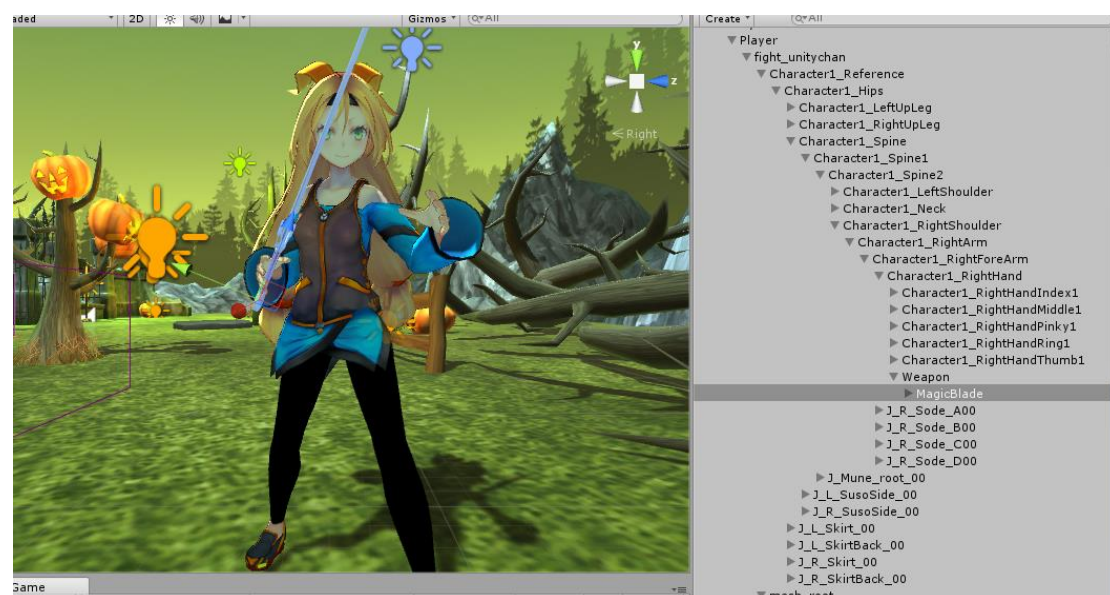
第三人称动作游戏视角，即需要将摄像机与人物绑定，并通过监听鼠标移动来改变观察角度。

我们提供了两种视角控制方式，分别是镜头自动跟踪人物前进方向的 MultiPurposeCamera 方式，以及能够根据鼠标控制镜头方向的 CameraMouseTurn。这两个镜头控制脚本都是 Unity SDK 提供的。



### 3.3 场景交互

场景交互也是该项目实现的另外一个难点，需要实现人物与怪物、NPC、场景物品等的交互功能。人物与 NPC、场景物品的交互实现相对较为简单，即发生交互时改变 NPC、物品的动画并添加上一些对话框等 UI 控件即可。至于人物与怪物的交互（主要是战斗）则存在许多细节需要仔细考虑其实现，比如如何检测人物打击范围，如何显示打击特效与打击过程中动画的变换。在这个问题的处理中，游戏引擎的使用依然是关键。



首先是打击碰撞检测。为了游戏性方面的考虑，人物最好能够使用多种不同类型、不同打击范围的武器，这样就不能仅仅通过相对距离来判断是否存在打击。我们目前考虑的解决



方案是把每个武器单独作为一个 Game Object 绑定在人物的 Object 上，并给每种武器画上不同的 Collider。当用户按下某个攻击键时，才检测武器的 Collider 与怪物的 Collider 是否有接触。

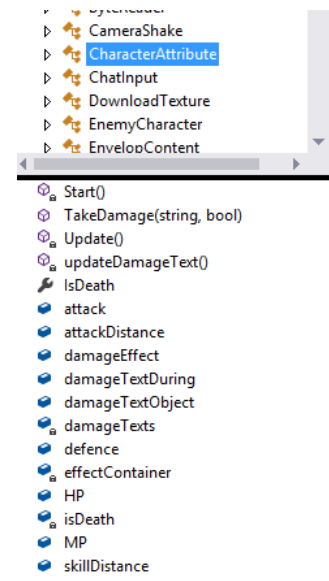
但在实际开发过程中，我们在检测是否有接触时，发现用 Collider 来判断近距离攻击的算法并不现实，效果十分不理想。因此后来我们想了另外一种判断方法，即基于人物和怪物的距离、人物朝向方向来判断：当人物和怪物距离在攻击范围内，且人物朝向怪物时（人物朝向 forward 向量点乘人物位置指向怪物位置的单位向量结果大于 0 则判定该攻击有效。

```
protected void WeaponAttackEnemies( ) {  
    foreach ( GameObject enemy in allEnemies ) {  
        EnemyCharacter e = enemy.GetComponent<EnemyCharacter>( );  
  
        float distance = Vector3.Distance(e.transform.position,  
transform.position);  
  
        Vector3 dir = ( e.transform.position - transform.position ).normalized;  
  
        float direction = Vector3.Dot(transform.forward, dir);  
  
        if ( direction > 0 && distance < attribute.attackDistance ) {  
            e.BeAttacked( );  
            //if ( !isShaking ) StartCoroutine(Shake(0.1f));  
        } else {  
            //e.BeAttacked(false);  
        }  
    }  
    return;  
}
```

### 3.4 角色状态维护

为了增加游戏的趣味性与挑战性，角色必须有诸如血量，攻击力，防御力等属性，且会随着游戏的进行而不断变化。实现这一功能并不困难，只需要定义一个包含所有这些信息的类即可。但值得一提的是，为了降低各个模块的耦合性，我们应该把角色的状态维护脚本与控制脚本分别单独实现，以便往后修改更新代码时能够快速找到相应的位置。

我们为此编写了一个类 `CharacterAttributes`，将怪物、人物的共同属性都放在该类中实现，以便能够增加代码的复用性。该类维护了 HP, MP 等人物和怪物可以共有的属性，以及扣血，显示伤害文字等共有方法。

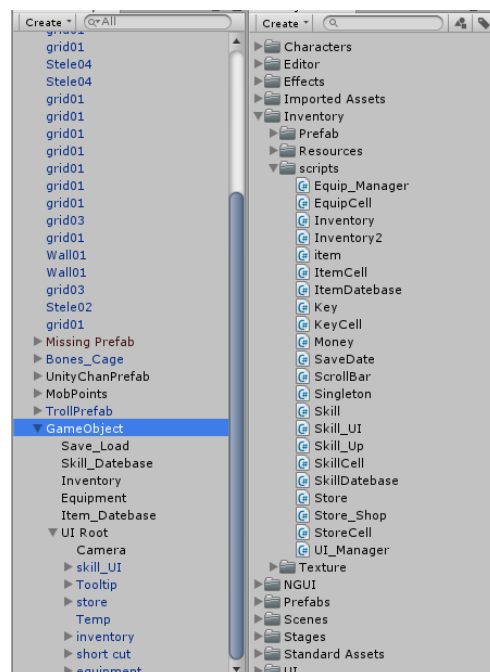


## 四、物品装备系统设计

### 4.1 物品系统设计

整个物品系统包括背包系统、装备系统和商店系统。所有系统都是以一个 prefab 为基础，将处理脚本挂到 prefab 上。我单独使用一个摄像机投影各种系统的 UI 界面，这样使得人物和场景的转换不会影响物品系统的操作。

整个物品系统的架构如下图所示：



系统一共设计了五个 GameObj，分别是：管理文件存储（游戏节点保存）的 Save\_Load 对象，管理技能数据的 Skill\_Database 对象，管理背包的 Inventory 对象，管理装备的 Equipment 对象以及所有管理物品信息的 Item\_Database 对象。每一个对象就是一个系统，由一个自行设计的 prefab 文件和一个或多个管理脚本组成。

以背包系统为例，首先需要设计一个 prefab，UI 设计如下：



UI 的设计主要是使用了 NGUI 插件，其中提供了一些成型的组件，比如各种材料组件等。基于这个 NGUI 插件的事件响应框架，设计了本游戏的物品系统 UI。

在背包系统中，每一个放置物品的位置是一个 cell，cell 需要设置碰撞检测，在拖拽物品是保证完整的放到 cell 中。UI 需要设计一个背景层，这里选择使用一个透明背景，组件在 NGUI 中提供。物品的图标来自 DOTA 的资源，背包下方的金币 UI 是通过 money prefab 实现的，分别设计一个金币图标，金币显示幕布和金币 text，然后连接一个 money 的脚本即可。除此之外还需要一个显示物品信息的 prefab，这个 prefab 只需要设置 UI Panel 的脚本，显示不同的内容即可。我将背包系统的 prefab 做层：一层背包格子图片、一层物品图片。分别用脚本控制物品的数量、物品的拖拽、物品的重复检查等等。



装备系统中也同样需要设计一个装备栏的 UI，与背包 UI 类似的，设置 6 个 cell，每个 cell 需要设计碰撞检测，对应的装备物品也需要碰撞检测。我将信息的显示栏封装成一个单独的模块，所以在背包系统和装备系统中这个部分是可以重复使用的，只需要检测物品的类型和信息，然后调用输出即可。



在物品和装备的基础上，我设计了一个商店，商店也是一个单独的 GameObj，由 prefab 和脚本构成。这里边功能脚本相对复杂一些，首先对于“购买”事件的响应，在获取用户交互信息之后，需要调用对应的修改函数，修改已有物品或者是添加物品到背包之中。所有的处理过程和函数我将其封装到一个 store 脚本中。



在 store 脚本中设置了监听事件，在“购买”事件发生了之后，需要调用金币处理函数，对应的更改金币的信息，然后将 UI 界面上的金币数量刷新。



## 4.2 技能系统与几点储存

技能系统相对于物品系统还是比较独立的一个模块，首先先在 Skill\_Database 模块中存储技能的所有信息，包括技能的名称，类型（主动、被动），说明以及序号。然后调用 tips 模块输出对于技能的说明，同时我单独设计了快捷键栏，对于每一个 key 值 1-5，z-b 都分别设计了一个 prefab，功能重点在 key 的脚本。在 key 的脚本中分别设计了初始化函数，对于技能的释放、信息存储、添加清空快捷键、以及技能的释放 CD 控制等。



在技能系统方面，UI 的架构设计是主要的问题，这里边我在 skill\_UIprefab 中绑定了控制脚本，所有的对于事件的响应处理函数都放在这个脚本中。首先初始化技能列表，然后根据用户的交互，监听点击事件，拖拽事件，分别调用处理函数。主要的处理函数有三个部分，一个是技能的修改，如技能升级，技能学习等；一个是技能的拖拽更新，即将技能拖放到快捷键栏中；还有一个是技能 CD 的处理，这个可以在 NGUI 中使用对应的接口以供调用的。系统处理的**难点在于窗口的控制，比如窗口、技能的拖拽、更新，隐藏与显示。比如在拖拽的过程中需要对之前的操作进行清空，还要能在隐藏窗口之后正确还原。**这些细节都是需要在设计时反复处理的，比较麻烦。



技能的释放 CD，我将功能封装到一个 CD 模块中，技能的 CD 可以调用 U3D 引擎中的 Time 类来具体设定。CD 的蒙版效果在引擎中有提供，可以直接调用，只要在脚本中设定就可以将蒙版效果与 CD 时间同步。

## 4.3 技术细节的实现

在 UI 界面的设计上，三个系统的界面设计尽管效果不尽相似，但是设计的思路是一样的。主要是 cell 的使用，其中要注意物品和 cell 的碰撞检测。prefab 的层次设计在背包系统中已有说明，这里不再赘述。下面主要介绍一下各个脚本的设计实现。

仍然以背包系统为例，首先需要在脚本中初始化背包列表，添加各种存储物品。物品信息需要从 item\_database 中提取，这里我简单的把数据库设计成了文件，把信息写到文件中去。初始化伪代码如下：

```
Initial {
    //添 count 个物品
    while(){
        inventory .Add(new item());
    }
    //从数据库中获取物品信息
    datebase =(find itembase)
    while(){
        AddItem [];
    }

    //初始化背包
    InitInventory ();
}
```

```

        draggingItem=false;
        draggedItem = new item ();
    }

```

背包系统处理物品是添加物品是需要反复用到的 ,而且在与商品系统的关联过程中也需要用到 ,所以最好的方式就是将其封装。添加物品的过程中需要判断原有物品中是否有本物品 , 如果没有则寻找一个空白 cell 添加进入 , 如果已经存在则需要在原有 cell 上更改物品数量。下面是添加物品的伪代码 :

```

AddItem(物品 id){
    while (遍历背包物品) {
        if(该 id 物品在背包内){
            if(背包有空格){
                if(inventory[i].itemNum < inventory[i].itemMaxNum){
                    物品数量加 1;
                }
            }
        }
        else (如果该 id 物品不在背包内){
            if(inventory[i].itemName==null){
                添加物品
            }
        }
    }

    if(如果没有找到未满的物品){
        while(遍历背包)
        {
            if(inventory[i].itemName==null){
                创建新物品
            }
        }
    }
}

```

类似的其他系统的控制脚本思路也大致相同 ,需要对每一个相应事件进行判断然后对应的处理。比如在装备系统的控制脚本中 ,同样是定义了初始化函数、更新装备函数、显示装备函数、保存和加载装备函数。而这些函数的调用全部都是通过 NGUI 的事件响应机制。

最后整个系统的 UI 管理界面我封装了一个上层的管理脚本 ,叫做 UI\_Manager ,它是被绑定到了 UI\_Root 下 ,也就是 UI 的根对象上 ,下面对这个管理脚本做一下详细分析 :



首先获取 UI 界面的所有对象，包括各种背包、技能框、信息栏等等。

```
public Transform equipment;//装备框
public Transform inventory;//背包框
public Transform short_cut;//快捷键框
public Transform skill;//技能框
public Transform store;//商店框
public Transform temp;//拖拽图标
public Transform tooltip;//信息框
public List<Transform> UI=new List<Transform>();
```

然后初始化这些 UI 的位置和顺序，以及事件：

调用 transform .FindChild()函数来获取各个组件，这与 Android 等很多语言都很类

似，就不多赘述了。之后调用 UI.Add 初始化 UI 的位置：

```
UI.Add (equipment);
UI.Add (inventory);
UI.Add (skill);
UI.Add (store);

//初始化 UI 顺序及 UI 事件
for(int i=0;i<UI.Count;i++){
    SetDepth(UI[i],i+1);
    UI_Event(UI[i]);
}
```

其中在 UI 设定的时候需要设置其深度，事件监听，UI 置顶等，这些也都以函数的

方式实现：

**//UI 单击事件**

```
void UI_Click(GameObject ui,bool isPress){
    if(Input .GetMouseButtonDown(0)){
        UI_Top (ui.transform .parent .parent);
    }
}
```

**//UI 置顶**

```
public void UI_Top(Transform ui){
    for(int i=0;i<UI .Count ;i++){
        if(UI[i]==ui){
            UI.Add(UI[i]);
            UI.RemoveAt(i);
            UIOrder ();
            break;
        }
    }
}
```

```
    }  
  }  
}
```

**//设置 UI 深度**

```
void SetDepth(Transform ui,int depth){  
    if(ui.name == "store" || ui.name == "skill_UI")  
        ui.Find("Win/Scroll View").GetComponent<UIPanel> ().depth = depth;  
    ui.Find("Win").GetComponent<UIPanel> ().depth = depth;  
}
```

## 五、技能特效设计

根据剧情设置，主人公提拉米苏的手持武器为剑，并且具有火系魔法，因此在技能特效的设计上主要分为魔法特效的设计与刀光效果的设计两部分。

### 5.1 魔法效果设计：

由于游戏中需要比较炫的特技效果，因此原本希望利用 Unity 自带的 Particle System 完成，但实际的各个种类粒子特效的效果比较单一，需要多种效果进行组合完成。

魔法效果的设计上主要以以下几种效果为主：

#### 火焰特效

由于剧情中主人公为一个火系魔法师，因此在特效的设计上最重要的一项特效便是火焰特效，剑所指之处可以生成一团火，对于敌人造成大幅伤害。

在火焰效果的实现上，主要采用了粒子系统中的如下几种单独的粒子特效



通过调节每个粒子特效的 Duration, Start lifetime, Start Size, Start Speed 等粒子的设定参数，并进行整合即可得到预期的效果。

最终效果如图：

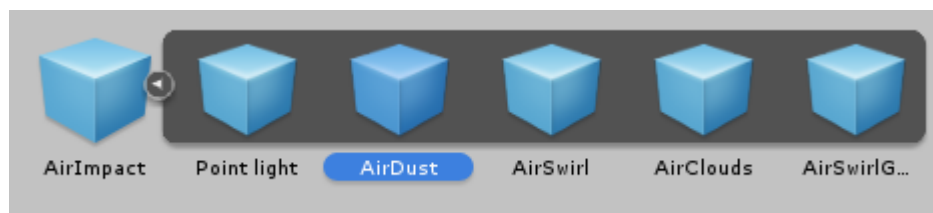


火焰特效

## 烟雾特效

在躲避敌人的过程中主人公还可以通过释放烟雾的障眼法方式进行短时间的缓冲,因此还设计了烟雾特效。烟雾特效主要模拟的是空气中的气体与灰尘的效果,起到阻碍视线的作用。

烟雾特效与火焰特效的原理类似,但主要的粒子系统使用以气体与烟尘的相关特效为主



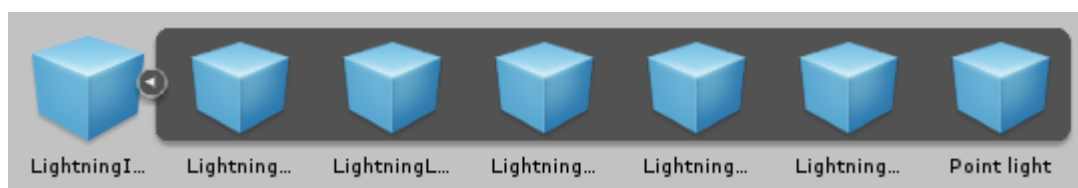


烟雾魔法特效

## 闪电光特效

与火系相伴的更加强有力的伤害效果便是电击 ,电击既可以瞬间减慢敌人的速度又可以远距离对敌人造成重创，是一种十分高效的技能方式。

同样与前两种特效的制作方式类似，将基础的 Partical 换为闪电相关进行组合。





闪电光特效

为了突出特效的炫的特点，在每个粒子系统的重心设置点光源，因此使得特效具有了魔法的属性，带有自发光的性质，对普通粒子系统特效平淡生硬的特点进行了一定的改善。

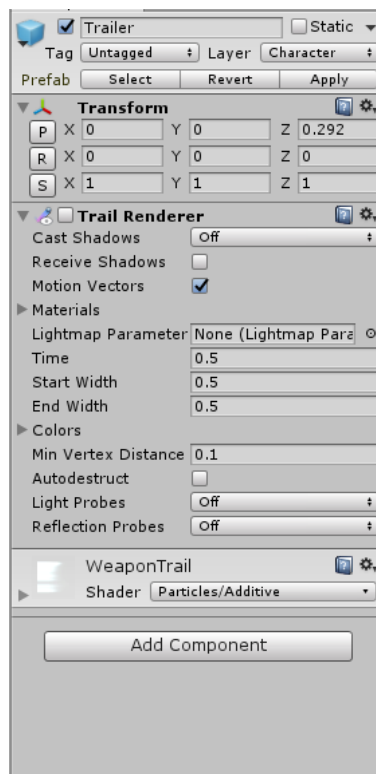
由于初次尝试 Unity，尝试几次自己组合 Partical System 的元素制作特效的过程中很难做出炫酷美观的效果，因此最后游戏的实现过程中借鉴了网上流传的几个特效成品，使得游戏的画质效果得到提升。

## 5.2 刀光效果设计

刀光效果的实现首先为 Weapon 添加一个 Trailer，追踪武器的运动轨迹，并对应设置预先找好的刀光贴图，即可沿着轨迹的路径制作刀光效果。

其中，对于 Trailer 的设置如下：





随动作的刀光实现效果

## 六、场景设计及搭建

根据游戏剧情的需要和 ARPG 游戏场景切换的需要，游戏场景主要搭建了两个场景：村庄场景和战斗场景。

村庄场景主要用于人物教学关卡的实现，人物装备强化和商品交易的需要以及人物剧情的触发等等。战斗场景则用于提供玩家操控的人物和 AI 战斗的场所。在战斗场景中会出现一定数量的“怪物”，当玩家将 AI 消灭以后，则提示玩家成功通过关卡，回到村庄来触发进入下一关卡的剧情。



村庄场景

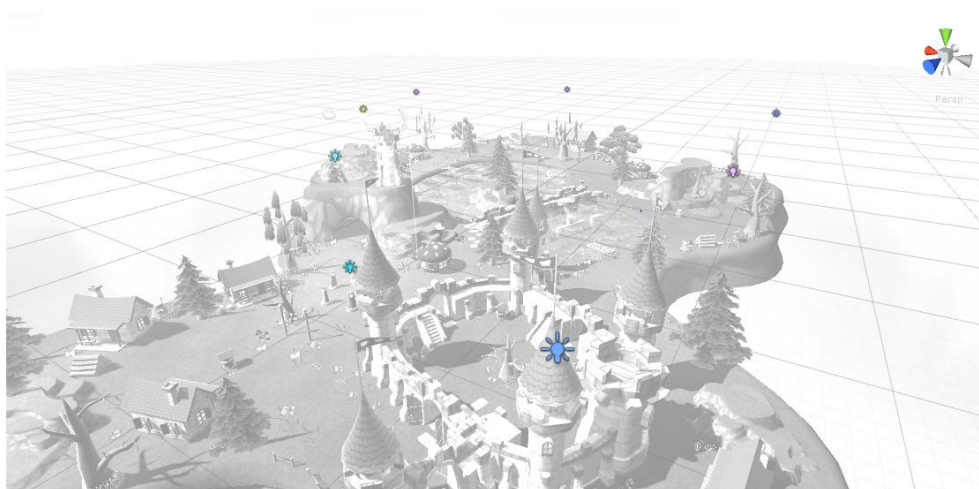


战斗场景



在游戏场景的实现过程中，场景的搭建是必不可少的。由于对于物体建模技术掌握的了解较少，所以在场景的实现过程中，绘制场景的资源主要从网站上下载和获取得到。将各个物体模型导入之后，剩下的工作便是拼接和搭建。但是搭建出场景只是场景绘制的第一个步骤。合理的光照和碰撞检测的设置对于游戏的体验性同样重要。

## 6.1 场景光照的设置：



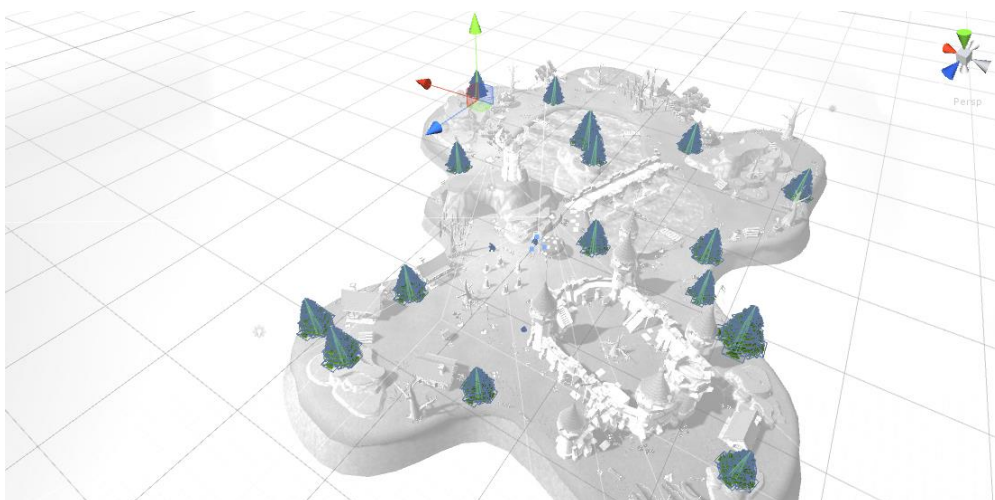
场景中的光源的设置主要通过一个平行光源和数个点光源覆盖实现。平行光源（方向光）用于显示游戏中阳光或者主要光源的方向，同时也是用于显示阴影的主光源。而分布于场景各处的点光源可以填充场景中的黑暗和阴影区域，同时将场景细节和各个地方的景物显得更加明亮。补充光的光线可以表现出景深和逼真的感觉。



通过场景中各个光源的交互和叠加，最终达到想要实现的主体或者情景。期中关于场景主体色的选取和绘制，只需要在添加光源时选取好光源的颜色即可以实现不同的主题色的设定。同时在绘制阴影时没有必要将场景中的每个光源都绘制阴影，而且绘制阴影的代价是十分昂贵的（尤其是光线跟踪阴影的情况）。在我们游戏的场景中只需要为主要场景设置阴影即可。



## 6.2 碰撞检测的设定



为增加游戏场景的真实性，游戏模型的碰撞检测的设定是必不可少的。在碰撞检测设置的过程中主要是通过 mesh collider 设定，即通过物体的 mesh render 相连顶点得到得到物体的 collider。但是 mesh collider 的方法只对于凸多面体适用，而且为了减少计算，

Unity 在生成物体的 collider 时会自动选取其突出的顶点相连。而如果物体具有空洞或者伸展的地方较多，用 mesh collider 的方法则会出现空气墙。



例如上图中城门的 collider 就无法采用 mesh collider 绘制，否则会出现空气墙导致人物无法穿过的情况。所以在添加城门的 collider 时，采用了 box collider 堆叠的方式添加。

为增强游戏的趣味性和真实性，而且考虑到我们是 ARPG 实时战斗类型的游戏，在场景中几乎要对场景中的每个物体添加碰撞检测的 collider。

## 6.3 场景小地图的添加和绘制

为增强游戏的可玩性和趣味性，根据玩家游戏体验的需要，在游戏中我们增添了小地图的显示功能。

具体的方法是在玩家的头顶上方放置一个方向向下的摄像机，将这个摄像机放置在人物的组下面，这样这台摄像机的方向和位置会根据人物的位置和方向变化而变化，但是由于人物具有跳跃的动作，但是需要保证摄像机的 y 轴坐标不会根据人物的 y 坐标变化而变化，控制小地图摄像机的 y 坐标不变需要为摄像机额外添加控制显示的脚本。



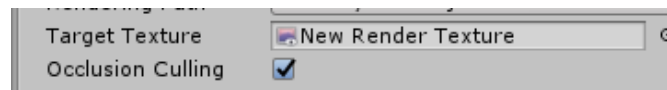
```

1 using UnityEngine;
2 using System.Collections;
3
4 public class SmallMap : MonoBehaviour {
5
6     private Transform target;
7
8     private float preserveY;
9     // Use this for initialization
10    void Start () {
11        target = GameObject.FindGameObjectWithTag("Player").transform;
12        preserveY = this.transform.position.y;
13    }
14
15    // Update is called once per frame
16    void Update () {
17        //Debug.Log("Update");
18        this.transform.position = new Vector3(target.position.x, preserveY, target.position.z);
19    }
20 }
21

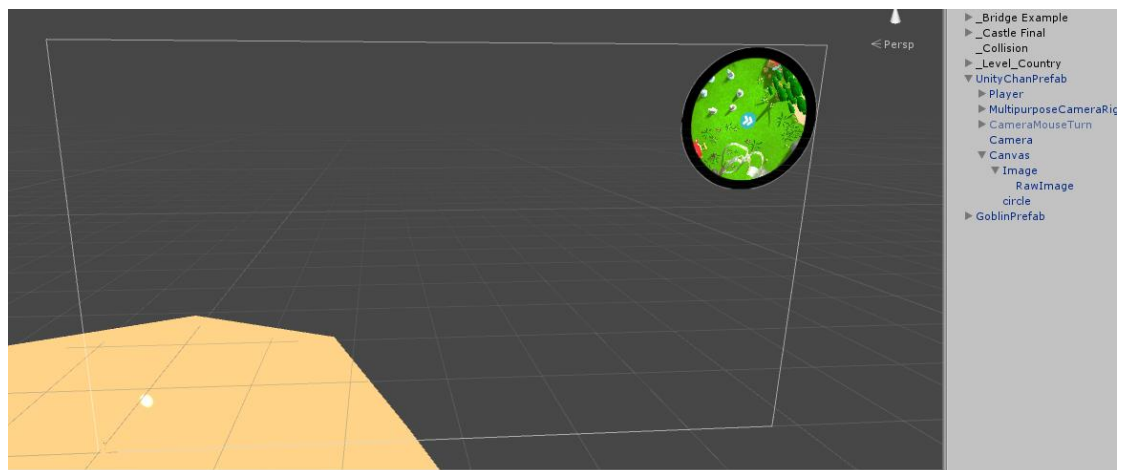
```

这样可以控制摄像机的位置只随着人物在 xz 平面的位置变化而不会根据人物的跳起作用而变化，同样摄像机的方向也不会根据人物的旋转而改变。

摄像机的位置设置好之后需要添加设置一个 texture，然后将摄像机的 target texture 设置为相应的 texture。这样摄像机拍摄下来的影像就会映射到相应的 texture 上。



得到人物头顶摄像机之后需要将摄像机拍摄出来的图像显示到我们的最终的视口上，接下来需要添加一个 canvas。



然后在 canvas 中添加一个图片 ( image )，然后将它的 texture 设置为我们摄像机映射的 texture，这样在这个图片中显示的就是人物头顶摄像机拍摄得到的影像。为了区分小地图的影像和真实地图的图像，我们给小地图添加了边框的效果，并让小地图显示限制在一个

圆形区域中。



主要方法是将 RawImage 放在另一个 image 中，而它的上一层 image 则是一个圆形的图片用来限定 image 的显示范围。同时 circle 则是一个边框，与 image 放在同一层，用来显示小地图的边框效果。这样小地图初步的效果已经实现了。

为了突出显示下地图人物的位置和方向，并且便于和地图中的其他 object 分开显示，同样在人物上方添加一个图片。



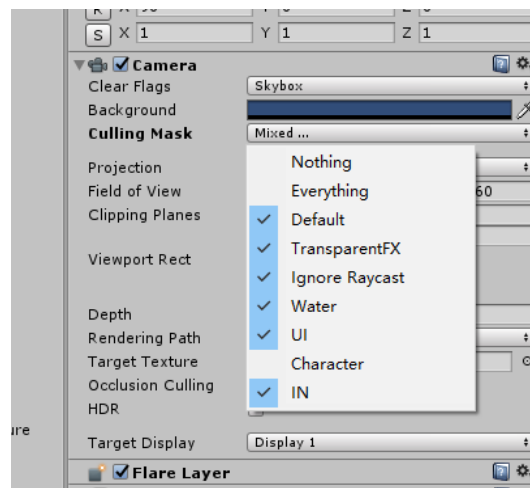
同样为图片添加一个控制显示的脚本，保证图片只是根据人物在 xz 平面上的位置和方向变化，而 y 坐标不会根据人物的跳起而发生改变。

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class arrow_height : MonoBehaviour {
5
6     private Transform target;
7     private float preserveY;
8
9     // Use this for initialization
10    void Start () {
11        target = GameObject.FindGameObjectWithTag("Player").transform;
12        preserveY = this.transform.position.y;
13    }
14
15    // Update is called once per frame
16    void Update () {
17        this.transform.position = new Vector3(target.position.x, preserveY, target.position.z);
18    }
19 }

```

设置完成之后，为保证主摄像机不会看到添加的标识人物位置的图片显示，而头顶摄像机不会看到人物，需要将标记图片和人物设置在不同的 level ( 层 )，摄像机的显示层同样设置显示不同的层级，这样可以保证每个摄像机看到的只是它们需要看到的 object。



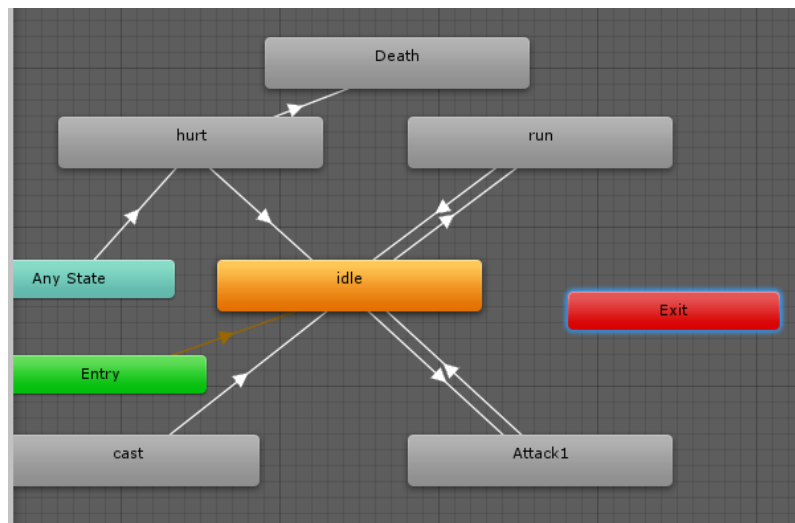
摄像机层级显示的设置在 Culling Mask 属性中，例如在头顶摄像机中不显示 Character ( 人物层 )，同时将人物设置在 Character 层，这样在头顶摄像机便不会看到人物，而只会看到我们设置的图标。



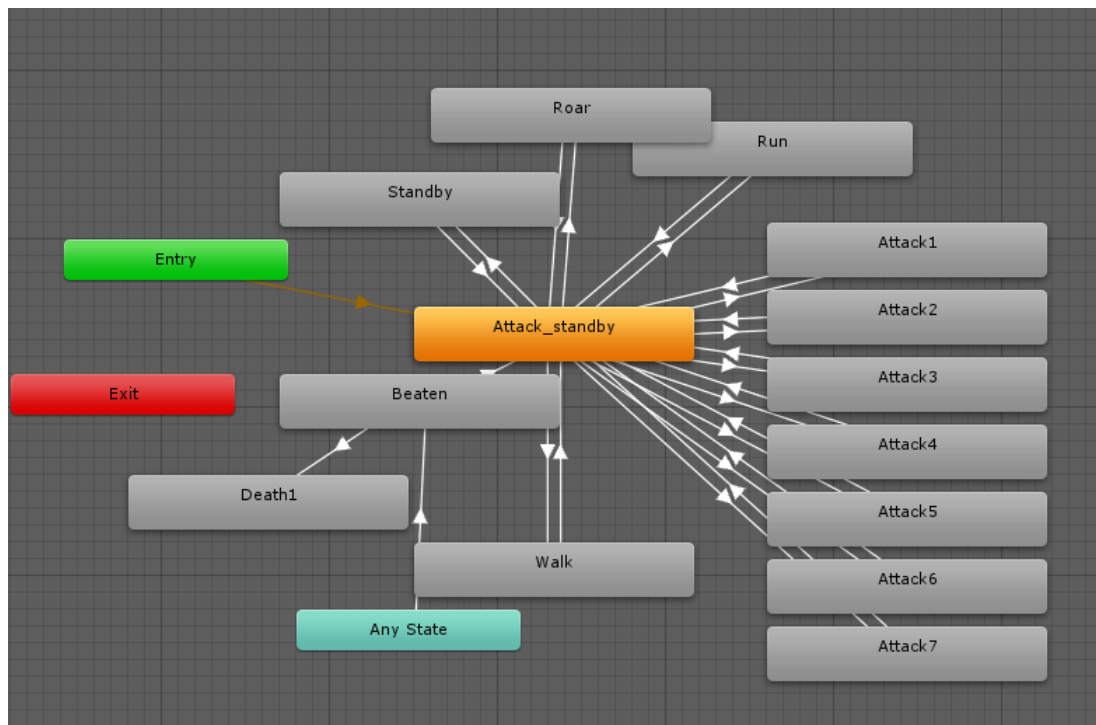
到此小地图的添加以及绘制已基本完成。

## 七、怪物 AI

在动作游戏中怪物的 AI 是玩家最为看重的一点。在设计怪物 AI 之前首先要实现怪物的动画状态机。目前我们游戏中只加入了两种怪物，因此我们分别为两种怪物设计了动画状态机：



哥布林动画状态机



Boss 动画状态机

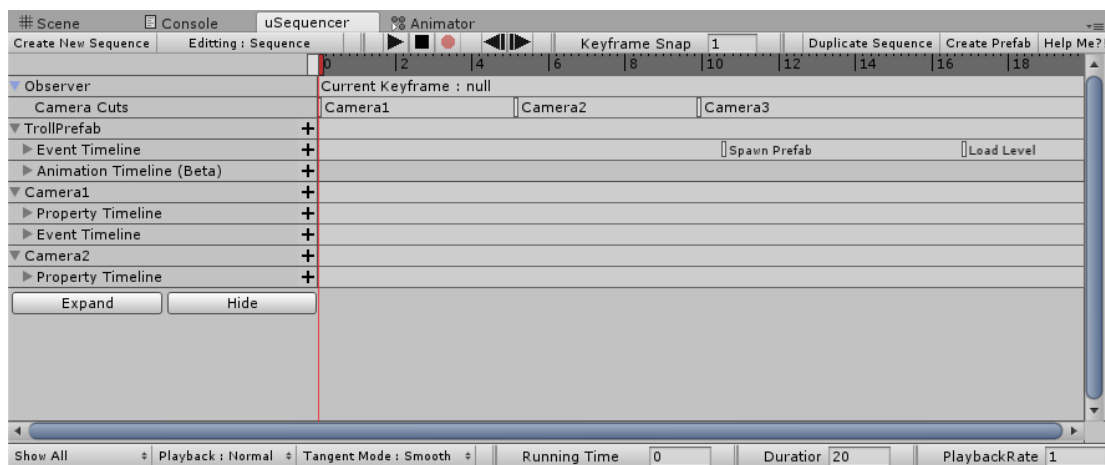


怪物动画状态机的控制与人物状态机看似没什么太大区别，但人物状态是玩家控制的，而怪物状态则完全是由程序控制的。我们设计的怪物有三种状态：巡逻，追赶玩家和战斗。

当怪物距离玩家较远时，怪物在固定的 3 个点来回缓慢行走 ( Walk ) 同时关注自己周围是否有玩家进入；当怪物发现有玩家进入周围时，进入追赶玩家状态，朝着玩家的方向奔跑 ( Run )；在追赶状态，当玩家跑远了，怪物发现玩家已经不在自己范围内时则返回巡逻状态；反之当追赶上玩家，即玩家在自己攻击范围内时则进入攻击状态，此时不再移动，而是朝向玩家播放攻击动画。将上述过程转成代码实现如下：

```
if ( InRange(attribute.attackDistance) && attackTimer < 0 ) {           // attack state
    navAgent.Stop( );
    attackTimer = AttackInterval;
    animator.SetBool("run", false);
    animator.SetBool("attack" + Random.Range(1, attackNumber + 1).ToString( ), true);
} else if ( InRange(searchRange) ) {                                     // Found
    navAgent.Resume( );
    navAgent.SetDestination(player.gameObject.transform.position);
    animator.SetBool("run", true);
} else if ( navAgent.remainingDistance < 1 && mobPoints != null && mobPoints.Length >
0) {
    // arrive at one patrol point
    navAgent.Resume( );
    mobPointIndex = ( mobPointIndex + 1 ) % mobPoints.Length;
    navAgent.SetDestination(mobPoints[mobPointIndex].transform.position);
    animator.SetBool("run", true);
}
```

## 八、过场动画



现在几乎所有游戏开场过程中，通常都会有一段唯美的 CG 动画作为游戏的背景故事介绍。虽然在游戏过程中没法用 CG 做剧情动画，但如果能用现有游戏资源实时播放一段剧情，必要时还可以让玩家的角色参与其中也是一种很不错的选择。

对于一段完整的剧情动画通常需要包括：镜头切换、各种动画、炫丽特效、悦耳的声音、旁白字幕、交互 UI 等等。在本项目中我们剧情动画的实现主要使用 Unity 的一个插件——USequencer 来实现。上图为 USequencer 的编辑界面图，与一般的视频编辑软件十分类似。

属性时间线是用来直接控制游戏物体的所有属性，可以使用动画曲线来控制所有东西。这点跟 Animation 动画一样，也是使用曲线和关键帧的方式控制每个角色(GameObject)的最基本属性：Transform、Material、Camera、Renderer 等等；事件时间线用来添加和执行各种事件，是我们使用最多的时间线。在 USequencer 中已经提供了 100 多个不同种类的事件，我们可以直接使用这些事件完成很多的逻辑控制。事件时间线是一个万能的时间线，可以控制任何东西，但需要你自已花时间去开发它的无限可能。动画时间线是作者开发用来支持编辑、预览和播放 Mecanim 动画的时间线，只能用来控制 Mecanim 动画。目前

这个还是 Beta 版，功能不是很稳定，所以通常会使用时间线的一些功能来替代它。路径时间线允许你把一个物体沿着一个路径运动，通常用于相机的轨迹运动。

此外 USequencer 一个最强大的特点就是可以一边编辑剧情，一边预览剧情。只需拖动工作面板上时间轴上的小滑块就可以预览剧情的效果而又不需要 Play 游戏，可以做到对动画、特效和事件逐帧预览，来回反复预览等等，功能十分强大。

# 九、游戏剧情

## 1、世界观

泰雷诺亚大陆，分为三块。东部土地最为富饶，由希尔德帝国所统治。希尔德帝国，乃千年帝国，其初代皇帝为泰雷诺亚大陆上唯一一位成圣的魔剑士，当年一人横扫各王国，威震八方，其威名千年而不衰，甚至据传他仍然活到现在。西部土地略微贫瘠，资源的匮乏使得战火纷飞，各路军阀组织成为联邦，遍布于广袤的大陆间，时而交战。而横贯东西的是巨大的无尽海，范围之广让人不敢想象，而它的最中间有被人称之为“十大绝境之一”的无尽深渊，没人知道这片海域的深度有多少，并常伴有狂风大浪，进入者九死一生。这也就形成了天然的一壁垒，横跨在东西大陆之间。

泰雷诺亚大陆，是被神所庇佑的大陆。诸天众神有三十六，各司其职，庇护泰雷诺亚大陆的为火神赫尔，火神赋予大陆上的生灵力量，每一位出生在泰雷诺亚大陆上的人类，亚人类，都将获得火神的印记在其 12 岁之日，在这之后，人类就可以与散布于整个世界的火之元素所沟通，使用精神力操纵火元素。当然，火神并不是青睐于每一个人类的，每个人类都有一项直观重要的属性，叫火元素亲和力，火元素亲和力高的人才可以更好地调动天地间的火之力。每一个到了 12 岁的孩子，都必须前往神殿参加火元素亲和力的测试，以判断她未来所从事的事业，火元素亲和力较高者，以为成为一名光荣的试炼者。

试炼者乃一种特殊的职业，试炼者通过不断习武，并且吸收火元素晶核来提升自己的操控火元素的能力。火元素晶核，是一种结晶物质，分布于泰雷诺亚大陆，可以从矿脉中获取，也可以从泰雷诺亚大陆的妖兽体内中获取。试炼者又分为数种，可以是一名剑士，将火之力灌注到长剑之中发挥出更强的力量，又或者是一名魔导士，通过操纵空气中火元素来杀伤对手。然而，最为稀有的是魔剑士，她魔武双修，将火元素魔法最为斗气外放，威力极大。成

为一名魔剑士需要极大的元素亲和力，史上少有人能够达到这样的要求，最近一位伟大的魔剑士便是希尔德帝国的开国皇帝——希尔德一世。每一位试炼者的人类都将踏上试炼的征程，通过不断击杀散布在世间的魔兽以磨砺自身，并且获得直接来自火神赫尔的嘉奖。

## 2、故事背景

故事发生在希尔德帝国的漠城内，泰雷诺亚历 1681 年。漠城，是边境城市，地处希尔德帝国与北方大漠的交界处，北方大漠一片荒凉，唯有少数蛮族存在。而莱卡家族，为城市内中赫赫有名的大家族，在这座城市已经称霸百年了。莱卡伯爵，也就是莱卡家族长，即为该城的城主，是一位大剑士，驻守该城。

虽然是边境城市，但是因为大漠荒凉，几乎不可能有巨大的灾难，所以该地驻兵稀少。但是，谁也没想到的是，某一天在荒芜的大漠之中，悄然打开了一个巨大的时空裂缝，来自其他大陆的人类，纷纷从裂缝中涌出，有来自被水神蒂亚庇护的大陆，有来自被土神震旦庇护的大陆等等，数以万计的不同人类集结于大漠之中。这是暴风雨来临的前兆。就在那一天，乌云笼罩了漠城，遮蔽了天空，但那并不是乌云，而是成百上千台魔导飞行器摆成的阵列。漠城，毫无意外地，沦陷了。莱卡家族，漠城全城之人，在这场巨大的灾难面前，几乎无人幸免。除了，一个女婴，名叫提拉米苏·莱卡。莱卡伯爵临近绝望之际，捏碎了一张他好友大魔导师菲特雷所赐予的传送符，将他唯一的女儿破开空间传送了出去，一同随他女儿出去的是一封信，信中是这么对菲特雷说道，希望菲特雷能够好好将他女儿抚养成人，并且不要妄图帮父亲报仇。

时间荏苒，转眼 10 年过去了，目光聚焦在一个偏僻的村庄内，在清澈的小溪边，有一个样貌可爱的女孩在小溪边玩耍，她身穿 XXX，头发 XXX，样貌 XXX(见模型)，那便是 12 岁的提拉米苏。12 岁的她已然是一名亭亭玉立的邻家少女，样貌可爱清纯。而这座被青山环绕的小村庄正式大魔导师菲特雷的隐居点。名义上，提拉米苏是菲特雷大魔导师的养女，

但她并不知道自己的身世，只知道自己是一名弃婴。本来，她可以就这样度过一生，但是在她 12 岁那年的火元素亲和力测试却改变了她的命运，她被测出具有史无前例的火元素的亲和力，完全有能力成为一名出色的魔剑士。

泰雷诺亚历 1691 年，希尔德帝国受到别的大陆的入侵，已经变得四分五裂。初代皇帝希尔德一世的复出仅仅维持住了进一步的入侵。希尔德帝国依然没落，只能维持原先腹部的残存区域，整个东大路的其余部分都被外族瓜分。菲特雷大魔导师得知了提拉米苏的资质之后，内心非常复杂，他知道如果希尔德帝国再出一个希尔德一世一样的人物，一定会重返当年的鼎盛，但是又想到当年莱卡家族家主的嘱托，他犹豫了。但是，在一次偶然的情况下，提拉米苏得知了她的身世。她毅然地决定踏上为父母亲报仇的这条路上，对抗来自其他大陆入侵的人类，寻找当年杀死自己双亲的凶手。就这样，她的传奇一生开始了。

### 3、主线剧情概述

#### 序章 特训

主要讲述提拉米苏在小村庄中受到了菲特雷大魔导师的训练，菲特雷大魔导师亲自教她关于魔法的相关知识，而同时又委托另一位剑术大师教她关于剑术的相关知识。

游戏中体现为教学关卡，使玩家掌握基本操作以及技能的释放。

#### 第一章 初出村庄

主要讲述提拉米苏在结束了她的初期训练之后，作为一名试炼者要走出小村庄进行试炼。她的第一个任务是由菲特雷大魔导师安排的，安排她参加一个商队的护送任务，护送商队穿过一片峡谷，期间有猛兽出没，提拉米苏就当担商队的护卫。同时商队的目的地是希尔德帝国的商业中心落雨城，在那里她将见到她父亲的老熟人狮鹫佣兵团长泰勒，据说可以得知她已故父亲的一些消息。

游戏体现为小怪的战斗，以及最终有一队劫匪的 boss 战。

## 第二章 约克堡

提拉米苏最终顺利见到了狮鹫佣兵团长泰勒，但是他也不知他父亲死于谁的手中，但他告诉提拉米苏，想要知道真相，必须自身强大起来。他推荐提拉米苏去约克堡，一所学习剑与魔法技能的学校，以她的资历，她一定能够成功入学。在哪里，她可以进一步磨炼自己的水平，学习战斗技能。并且，泰勒说，约克堡的学院长乃 10 年前大战的参与者，他应该知道些什么，关于提拉米苏的父母亲。

游戏体现为，加入势力，约克堡，并且可以学习新的技能。具体体现为多个小 boss 与不同的学生战斗，以及老师的试炼，最终可以有一个大 boss，才可以见到学院长。

## 第三章 应龙山脉

约克堡的磨砺已经不能满足提拉米苏了，她需要更多的外出试炼。约克堡院长也告诉他，想到知道父亲的真正信息，还需要前往前线，也就是现在的边境城市，应龙城。但是应龙城为军事重地，背靠应龙山脉，地势险峻。应龙山脉，传说中有龙曾存在，因此得名，内有大量魔兽的存在，难以横渡。而此次一去，提拉米苏必须经过应龙山脉，这将对她是一次巨大的挑战。她可以选择一人前往，或者是招募队友。

游戏体现为，开启招募系统，可以招募 NPC 帮手，或者是练级招募玩家一同完成挑战。整个应龙山脉分为数个小关卡，有不同的小怪和 boss，并且有大量的道具奖励。以及隐藏地图要素的存在。并且在应龙山脉中，可以寻找到提拉米苏父母亲的相关隐藏线索。

## 4、剧情详述

具体请见附件 游戏对话剧情

# 十、组员及分工

组员	学号	分工
庞博	3140103364	角色控制，战斗系统，项目整合
杨明亮	3140104873	技能、物品系统，界面设计
董泰佑	3140104431	场景设计，音乐设计
张文佳	3140104871	特效设计
胡义伟	3140105578	剧情设计



# 十一、附录

## 游戏对话剧情：

### 序章 特训

#### 梗概：

主要讲述提拉米苏在小村庄中受到了菲特雷大魔导师的训练,菲特雷大魔导师亲自教她关于魔法的相关知识,而同时又委托另一位剑术大师教她关于剑术的相关知识。

游戏中体现为教学关卡,使玩家掌握基本操作以及技能的释放。

提拉米苏从睡梦中醒来,出门遇见村民

提拉米苏：早上好,我的父亲呢？

村民：早上好,提拉米苏,今天又是个好天气,刚才遇见了菲特雷先生,他说他在村口的广场上等你。

提拉米苏：谢谢你,我这就去找他

找到菲特雷。

提拉米苏：父亲早上好,你怎么在这里？

菲特雷：早上好,提拉米苏,你没有睡糊涂吧,今天不是你最后一天的晨练吗？你都忘了？

提拉米苏：哦,好像是的啊,今天晚上就要出发护送商队了。

菲特雷：是啊,但是今天也不许偷懒,首先在我这练习魔法的技能,然后再找你的托勒密叔叔练习剑术的技能”

提拉米苏：好的，那我们这就开始吧

菲特雷：嗯，你准备好了吗”

提示：玩家按 enter 键开始法术练习，或者 esc 键跳过练习关卡。

菲特雷：首先是第一个技能：XXXX，你准备就绪了吗？集中精神！感受空气的元素波动，”

菲特雷：感受到了吗？准备了默念咒语，与元素沟通。1...2...3

提示玩家施法按键操作

菲特雷：来试试！

玩家若失败

菲特雷：再来一次，加油

菲特雷：好了，现在才练习一下这个新的技能，你还不太熟悉吧，提拉米苏

提拉米苏：“有点，不过我会尽力的”

菲特雷：“好，沉下心来，想象自己置身在元素的海洋中，与元素精灵沟通”

菲特雷：“来吧，试试”

提示玩家施法按键操作

玩家若失败

菲特雷：“再来一次”，“加油”

菲特雷：做的不错，今天也发挥得很好，下面我要去村里指导我的其他弟子学习魔法，你就

去找托勒密叔叔吧，他喝了一身酒气的回来，估计这会儿还在家里睡大觉，哈哈。快去吧，

不用担心会吵醒他。

提拉米苏：“嗯，我这就去”

提拉米苏：( 敲门 ) 托勒密叔叔，早上好，是我提拉米苏，快起床了”

托勒密：谁啊，一大早就鬼敲门，哟，是小提拉米苏，好好好，我马上来

托勒密：哎呀，怎么今天那么勤快，一早就来敲门了？是不是明天就走了，想叔叔了所以特地来看我？

提拉米苏：是啊，可想叔叔了。想和叔叔一起来练习剑术

托勒密：我就是知道你是来学习剑术的，进来吧

提拉米苏：好

托勒密：练习剑术必须做到形神合一，今天你的精气神还不错

托勒密：嗯，毕竟明天就要出发了，要养好精神啊

托勒密：好，那我们就开始吧。

展示基本攻击操作

托勒密：来，和我过两招

托勒密：不错啊

托勒密：这个动作还差点火候

托勒密：现在来练习下火元素剑气如何？集中精神，把剑气灌于武器之中，尝试一下

提示：释放剑术技能

玩家若失败

托勒密：“再来一次”，“加油”

托勒密：嗯，今天就暂时到此为止了，你学习的不错啊，没有枉费我教你 10 年剑术，你也

可以出师了，回去吧，快和你的朋友道个别，收拾收拾行李晚上就出发了。

：嗯，那我回去了，再见

晚上：

菲特雷：提拉米苏，在外面一定要小心啊，虽然护送商队并不危险，但是也要注意，善用我教你的魔法。

托勒密：嗯，你父亲说的很对，剑术练习也不要拉下了，路上小心。

菲特雷：父亲送你这一柄魔杖，这是千年古杉木制成的杖，威力极强

托勒密：你父亲都送你那么好的礼物了，我也不能落后啊，这是一把黑曜石打造成的单手剑，希望你用的顺手。

菲特雷：谢谢父亲，谢谢托勒密叔叔

菲特雷：嗯，走吧走吧，不然你托勒密叔叔又要舍不得你了。

托勒密：是啊，快走吧

提拉米苏：嗯，再见了。

## 第一章 初出村庄

梗概：

主要讲述提拉米苏在结束了她的初期训练之后，作为一名试炼者要走出小村庄进行试炼。她的第一个任务是由菲特雷大魔导师安排的，安排她参加一个商队的护送任务，护送商队穿过一片峡谷，期间有猛兽出没，提拉米苏就当担商队的护卫。同时商队的目的地是希尔德帝国的商业中心落雨城，在那里她将见到她父亲的老熟人狮鹫佣兵团长泰勒，据说可以得知她已

故父亲的一些消息。

游戏体现为小怪的战斗，以及最终有一队劫匪的 boss 战。

马车上

提拉米苏：这位小哥，这个商队是去哪里的呀？

小哥：唉，你这都不知道么？这个商队是落雨城中菲特家族的商队，听说里面装的是从无尽海那里运来的货物，价值连城。所以请了两位赫赫有名的帝国魔法师来护航呢。

提拉米苏：是吗？那我们这次岂不是会很安全？

小哥：当然，又这两位大师坐镇，谁还动了这个车队？

提拉米苏：嗯，有道理，难怪父亲让我来护送商队，第一次试炼应该不会很难的吧

小哥：小心了，马上就要穿过格尔蒙德大峡谷了，虽然不会有人类强盗来抢劫，但是猛兽可是不长眼的。

提拉米苏：嗯，是啊，会注意的。

小哥：敌袭敌袭！前方有 XXX 兽来了！

小哥：真不走运，刚说猛兽这就来了，准备好武器，快下车！

提拉米苏：好

提示：可以出现怪物信息，以及打怪的技巧提示

小哥：上啊。来决一死战！

胜利：

小哥：厉害啊，我们打退了怪物。

失败

小哥：喂，你还好吗？别这么死了！

小哥：唉，竟然会遇到怪兽袭击，我护卫了那么多次都难得见到一次。

提拉米苏：难道护卫不是挺危险吗？这种事不应该经常发生吗？

小哥：才不是。护卫大部分都是风平浪静的啊，要是次次这么危险谁还做这行？

提拉米苏：原来是这样。

快到峡谷口

提拉米苏：你有没有感受到四周突然很安静？

小哥：你这么一说好像是的。

提拉米苏：嗯，安静的有点出奇，有种危险的感觉。

小哥：你别吓我

路人：啊，后面有强盗（这人死了）

小哥：妈的，明明有两个魔法师坐镇那个混蛋眼瞎了来抢劫？

提拉米苏：看那个旗子！

小哥：完了完了，是风铃盗贼团，怎么遇上他们了！速速迎战！

提示：对人战斗信息

提拉米苏：来吧，你们这群盗贼

提拉米苏：不会让你们碰货物的！休想！

胜利：

小哥：厉害啊，我们击败了这群混蛋。

失败

小哥：喂，你还好吗？别这么死了！

进入 BOSS:

小哥：喂，你看，那是。。。那是风铃盗贼团副团长，希伯特！那可是大剑士啊！

提拉米苏：别愣着了，快逃，我们打不过的

希伯特：哼哼，你们逃不掉了，杀了我那么多手下，还妄图逃走？！痴人说梦！

提拉米苏：只有硬着头皮上了。一起上吧

小哥：怎么遇见你就这么倒霉呢？

提示：boss 战开始

当打到半血，或者撑过 N 秒后

小哥：不行，我撑不住了，提拉米苏，你还好吗？

提拉米苏：我还好，但是不能支援你。

小哥：唉，难道就要死在这里了么？



大魔法师：大胆！谁敢袭击我商队！

小哥：有救了有救了，大魔法师来了！

大魔法师：两位有劳了，我之前在车队前方对付敌人，来晚了实在抱歉！

提拉米苏：没事没事，您快去对付那人吧。

大魔法师：哈哈，区区大剑士，看招。

boss 被秒杀

大魔法师：这位小姐，刚才你抵挡了那个大剑士吗？

提拉米苏：。。

小哥：是的就是她，她实在是太厉害了。

大魔法师：不错不错，年轻有为啊，真是后生可畏！看你这样子不像是简单的护卫者啊。

提拉米苏：嗯，是的。我此去一行主要是来落雨城，见见狮鹫佣兵团长泰勒先生。

大魔法师：啊，太巧了，我和另一位同伴正好也是去找泰勒先生商量事情，如果不介意，等到了落雨城，能否一起同行？

提拉米苏：乐意之至。

大魔法师：好，那到了我们在城门口见面。

## 第二章 约克堡

梗概：

提拉米苏最终顺利见到了狮鹫佣兵团长泰勒，但是他也不知他父亲死于谁的手中，但他告诉

提拉米苏，想要知道真相，必须自身强大起来。他推荐提拉米苏去约克堡，一所学习剑与魔法技能的学校，以她的资历，她一定能够成功入学。在哪里，她可以进一步磨炼自己的水平，学习战斗技能。并且，泰勒说，约克堡的学院长乃 10 年前大战的参与者，他应该知道些什么，关于提拉米苏的父母亲。

游戏体现为，加入势力，约克堡，并且可以学习新的技能。具体体现为多个小 boss 与不同的学生战斗，以及老师的试炼，最终可以有一个大 boss，才可以见到学院长。

费尔金斯：这位小姐，久等了，先自我介绍下，我叫费尔金斯，和旁边这位一样都是宫廷魔法师，他叫霍尔，他可能比较不爱说话，但是是个好人。

霍尔：你好。

提拉米苏：你们好啊，我也没自我介绍呢，我叫提拉米苏。两位这是就要去见泰勒先生吗？

费尔金斯：是的。请随我们一起去吧，他在城中的佣兵团营地，在落雨城的东北角，那一大片都是佣兵团联盟的势力范围。而狮鹫佣兵团是那里最大的佣兵组织，泰勒团长可是无人不知啊。

提拉米苏：是吗？他那么有名？

费尔金斯：那是当然，他可是当年参与过 10 年前那场大战的勇士，被帝国授予勋章的男人。

提拉米苏：那么厉害，等不及要见一见这位先生了。

费尔金斯：哈哈，好，我们走吧。

费尔金斯：提拉米苏小姐，我们到了。

提拉米苏：好快。

费尔金斯：是啊，我们坐着风龙拉的马车当然很快了。

提拉米苏：哦，这样，风龙是什么？

费尔金斯：风龙是一种坐骑，盛产于应龙山脉。虽然叫风龙，但它实际上是地龙的一种，四肢非常健壮，适合长途跋涉。

提拉米苏：我也好想要这样一只坐骑啊。

费尔金斯：哈哈，会有这样一天的。

泰勒：哟，是谁在门口嚷嚷呢？

费尔金斯：泰勒！好久不见！

霍尔：嗯，你好泰勒。

泰勒：哈哈，你还是这么冷淡啊霍尔。

费尔金斯：他一直都这样。

泰勒：这位是？

提拉米苏：你好，我是提拉米苏。是我父亲菲特雷让我来找你的。

费尔金斯：菲特雷！你说的菲特雷大魔导师吗？原来你是他的女儿！失敬失敬！

霍尔：菲特雷！

泰勒：哦，你就是提拉米苏。你父亲跟我提过你，快快有请。

提拉米苏：好的，泰勒先生。

泰勒：就叫我泰勒叔叔吧，我与菲特雷也是旧识了。

提拉米苏：嗯，好，泰勒叔叔。

进屋坐下：

泰勒：小提拉米苏，你找我有什么事吗？

提拉米苏：泰勒叔叔，我想知道我亲身父亲的事情。据说您参加过当年的大战，知道发生在漠城的一些事。我想向您打听一下。

泰勒：你的亲身父亲？不是菲特雷吗？

提拉米苏：是的，我的亲生父亲应该是当年坐镇漠城的莱卡伯爵。

泰勒：莱卡伯爵？！这个名字好熟悉。哦，是那位莱卡大剑师吗？当年他一人可是面对数万敌人面不改色，真正地漠城同生同死的帝国伟大的勇士。可惜。。

提拉米苏：可惜？

泰勒：可惜漠城无人生还。

提拉米苏：您当时在那里吗？

泰勒：不，怎么可能，我是后来响应战争动员才加入战争的。

提拉米苏：这样啊，那您不知道我亲生父亲的一点消息吗？

泰勒：恕我无能为力。不过有一个人说不定可以。

提拉米苏：谁啊？快说啊叔叔。

泰勒：约克堡的学院长，劳伦斯。

提拉米苏：约克堡？

泰勒：哦，你还不知道。约克堡是落雨城最大的学院，学院你知道吗？就是很多人一起学习魔法和剑术的地方。

提拉米苏：嗯，第一听说。

泰勒：劳伦斯是目前那里的院长，据说他当年奋战在对抗异族的第一线，也是最初参与大战而幸存的一批人之一。

提拉米苏：他有可能知道我父亲的消息吗？

泰勒：是的。不过你得先进入约克堡。我想以你的资质，进入那里应该轻而易举，毕竟是菲特雷的女儿啊。哈哈。

提拉米苏：没有没有，叔叔过奖了。

泰勒：不用谦虚，刚才那位魔法师还对你赞不绝口呢。而且你进入约克堡，在那里你可以学习新的魔法和技能开拓自己的眼界。

提拉米苏：嗯，知道了叔叔，那我如何进入约克堡呢？

泰勒：你来的时间正好。这周末就是约克堡三月一次的选拔比赛，只要你通过的比赛，就可以进入约克堡。这样，我等会派人去取下比赛的手册，你可以仔细研究下。

提拉米苏：谢谢叔叔。

获得手册。查看比赛安排。

比赛三局，获胜二局即可。

提拉米苏：好多人啊，我该怎么报名呢？

提拉米苏：你好，我要报名参加约克堡的选拔比赛。

工作人员：好的，请问你是什么职业呢？剑士还是法师？

提拉米苏：我。。。我是魔剑士。

工作人员：魔剑士？！我的天哪？你是认真的吗？

提拉米苏：是的。

（骚动）

工作人员：如果是魔剑士的话，千年难遇，约克堡有规定，可以直接入学。但是。。。

提拉米苏：但是？

工作人员：但是还从来没有一个魔剑士加入过约克堡。。。我们需要验证一下你是否真的是魔剑士。

提拉米苏：好吧。

工作人员：刚才讨论了下，你将会被安排额外的比赛。你愿意吗？

提拉米苏：什么比赛？

工作人员：你将和两位约克堡学生比赛，以及一位老师。当然你不必取得胜利，只要你发挥出魔剑士应有的能力即可。

提拉米苏：可以。那我们就快开始吧。

工作人员：对了，如果你能获得胜利的话，学院将会给你额外的特殊奖励，这是院长刚刚决定的。

提拉米苏：劳伦斯先生？

工作人员：是的，我也很奇怪，不过你更要加油了。

提拉米苏：好的，一定努力。

比赛三局，赢一局获得奖励，三局获得更大奖励。

向导：恭喜你通过了约克堡的入学测试，魔剑士小姐。

提拉米苏：你好，我叫提拉米苏。

向导：哦，抱歉了，提拉米苏小姐，我是您在学院的向导，有什么事都可以问我，我随时给你答复。这是给您的约克堡学员徽章。请您收货。（叮咚，加入势力：约克堡）

提拉米苏：好的，会注意保管的。

向导：嗯，这里还有有关约克堡的各种信息，如果不明白了随时叫我。您现在可以在约克堡

四处逛逛都可以。

四处转悠，学习技能之类的，买道具武器防具。。

提拉米苏：唉，这里是学院长的办公楼吗？可以找到劳伦斯先生吗？

疯子：站住！这里闲人免入！

提拉米苏：我只是来找院长先生的。

疯子：院长先生可不是谁想见就见的。

提拉米苏：那我该如何见他呢？

疯子：你？没门！

提拉米苏：你怎么这么说话呢？

疯子：想见他，除非过了我这关！

提拉米苏：=。=（选择进入战斗）

成功

疯子：哼哼，看来你还有几份能耐，姑且让你过去。

失败

疯子：小丫头，再练几年再来吧。

提拉米苏：那人到底是谁啊？怎么还拦着人不让进去？好烦人啊

劳伦斯：哈哈哈哈哈，不要介意，他是莫雷，一个武痴而已。



提拉米苏：谁？

劳伦斯：你是在说我吗？

提拉米苏：嗯？

劳伦斯：哈哈，不逗你了，我早知道你要来了，提拉米苏小姐。

提拉米苏：你是？劳伦斯先生？

劳伦斯：正是。我已知道你的来历的。来，有请。

提拉米苏：谢谢，劳伦斯先生。

### 第三章 应龙山脉

**梗概：**

约克堡的磨砺已经不能满足提拉米苏了，她需要更多的外出试炼。约克堡院长也告诉他，想到知道父亲的真正信息，还需要前往前线，也就是现在的边境城市，应龙城。但是应龙城为军事重地，背靠应龙山脉，地势险峻。应龙山脉，传说中有龙曾存在，因此得名，内有大量魔兽的存在，难以横渡。而此次一去，提拉米苏必须经过应龙山脉，这将对她是一次巨大的挑战。她可以选择一人前往，或者是招募队友。

游戏体现为，开启招募系统，可以招募 NPC 帮手，或者是练级招募玩家一同完成挑战。整个应龙山脉分为数个小关卡，有不同的小怪和 boss，并且有大量的道具奖励。以及隐藏地图要素的存在。并且在应龙山脉中，可以寻找到提拉米苏父母亲的相关隐藏线索。

劳伦斯：你好啊，小提拉米苏，早就听说过你，唉，其实你三岁的时候我还抱过你呢

提拉米苏：惊！

劳伦斯：哈哈，不用那么惊讶，我和我的父亲，你的亲生父亲，莱卡是老朋友了。

提拉米苏：您认识我的亲生父亲？

劳伦斯：当然。我和他可是交情颇深。我和他当年在一所学院里面一起毕业的。是同级生。

提拉米苏：约克堡吗？

劳伦斯：不不不，约克堡是新建的一所学院。唉，那所学院已经不在，消失在了战火中。

提拉米苏：不好意思，不该问这个的。

劳伦斯：哈哈，没事没事。都是陈年旧事了。来，说说你父亲莱卡的事吧。你想问什么？

（未完待续）