



# **Teamwork four**

# **2021**

# Contents

**01.** 开发计划  
Development Plan

---

**02.** 项目规划  
Project plan

---

**03.** 系统设计  
System Design

---

**04.** 数据库设计  
Database Design

---

**05.** 类图与安全  
Class diagram and security

---

# 01.开发计划

01. Development Plan

---

# 1.1开发计划

## 1.1 Development Plan

时间	目标
第1周 (4.25-5.1)	实现主角前进、跳跃、冲刺、攻击
	实现怪物状态设计
	实现音频管理
	场景ui编写
	静态数据准备
第2周 (5.2-5.8)	绘制图片
	实现主角技能
	实现怪物状态切换
	实现主角动画
	游戏窗口ui编写
第3周 (5.9-5.15)	背包、地图、存档的实现
	场景搭建
	主角动画接口的调用，动画优化，音频接口调用
	怪物动画接口的调用，动画优化，音频接口调用
	实现怪物动画
第4周 (5.16-5.22)	场景搭建
	脚本与场景对接
第5周 (5.23-5.29)	项目测试，对各个接口调试，测试
第6周 (5.30-6.5)	添加游戏特效，完善游戏细节，数值调整
第7周 (6.6-6.12)	报告，开发结束

## 1.2 Process overview



# α版本

# β版本

完成

## 规划1

将项目的设计落地，继续完善设计类图，并在正式开始开发之前设计好系统和数据库、制定好计划和分工；磨刀不误砍柴工，希望大家重视这次作业，为下一次的团队Alpha冲刺作业提前做好准备；

## 规划2

计划阶段**4天**，Alpha冲刺阶段共计**10天**，冲刺阶段要求每天进行小组会议，讨论当前项目昨日进展、存在问题、今日安排（安排需要组长根据情况灵活修改）

## 规划3

## 规划4

# 02.项目规划

02.Project Plan

---

## 2.1 团队简介

### 2.1 Team Introduction



脚本人员

王国志

负责普通怪物移动方式的脚本开发，怪物攻击方式的脚本开发以及怪物AI的编写。



脚本人员

宋方泽

负责精英怪物移动方式的脚本开发，怪物攻击方式的脚本开发以及怪物AI的编写。



脚本人员

司维维

负责BOSS移动方式的脚本开发，怪物攻击方式的脚本开发以及怪物AI的编写。

## 2.1 团队简介

### 2.1 Team Introduction



**脚本人员**

**王振南**

负责从序列化文件中获取存档、相关特效的脚本开发。



**脚本人员**

**吴特锦**

负责从json文件中获取对话信息、背包内容、地图的脚本开发。



**脚本人员**

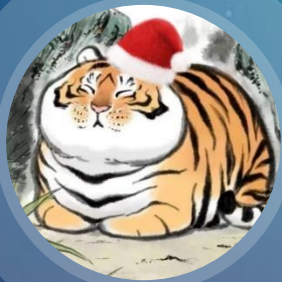
**汪碧桢**

负责UI界面上各种按钮以及设置界面里面的各种操作的脚本开发工作。



## 2.1 团队简介

### 2.1 Team Introduction



**脚本人员**

**宋喜仁**

负责音频管理、动画的  
脚本操作



**美工**

**毛富林**

负责怪物设计、UI界面  
设计、关卡设计、剧情  
设计、场景搭建



**美工**

**周美婷**

负责主角设计、UI界面  
设计、关卡设计、场景  
搭建

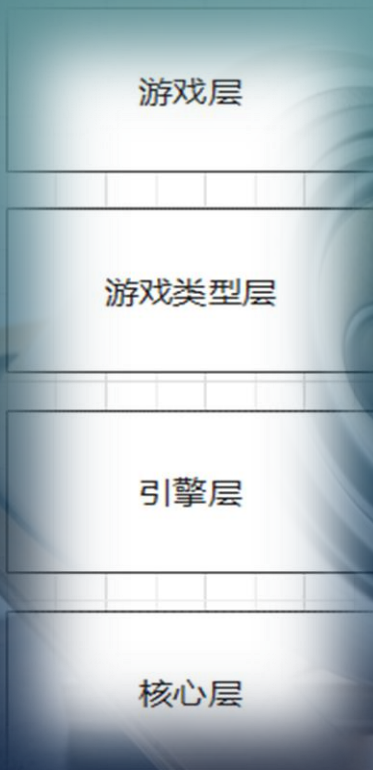
# 03.系统设计

03. System Design

---

# 3.1 体系结构设计

## 3.1 Architecture design



# 3.1 体系结构设计

## 3.1 Architecture design

### 信息层

这部分主要分为两个设计：一个是知识池（**Knowledge Pool**），另一个就是感知器（**Sensor**）。

知识池（**Knowledge Pool**）：将世界的数据信息搜集起来，通过**key-value**的方法存储来；感知器（**Sensor**）：即获取数据的方法，可以设置一个感知器来获取地图的信息并将信息存入到知识池中；

### 行为层

该层的作用就是用来处理上层的策略，这层的实现同样使用了行为树、状态机，比如在释放一个技能时，我们需要先集气，然后再释放，就可以通过行为树来实现。

在这一层中，会产生一系列的输出，有特效，有动画，还有声音等，然后就是接受运动信息，这部分包括空间信息（位移）和动画信息。

### 决策层

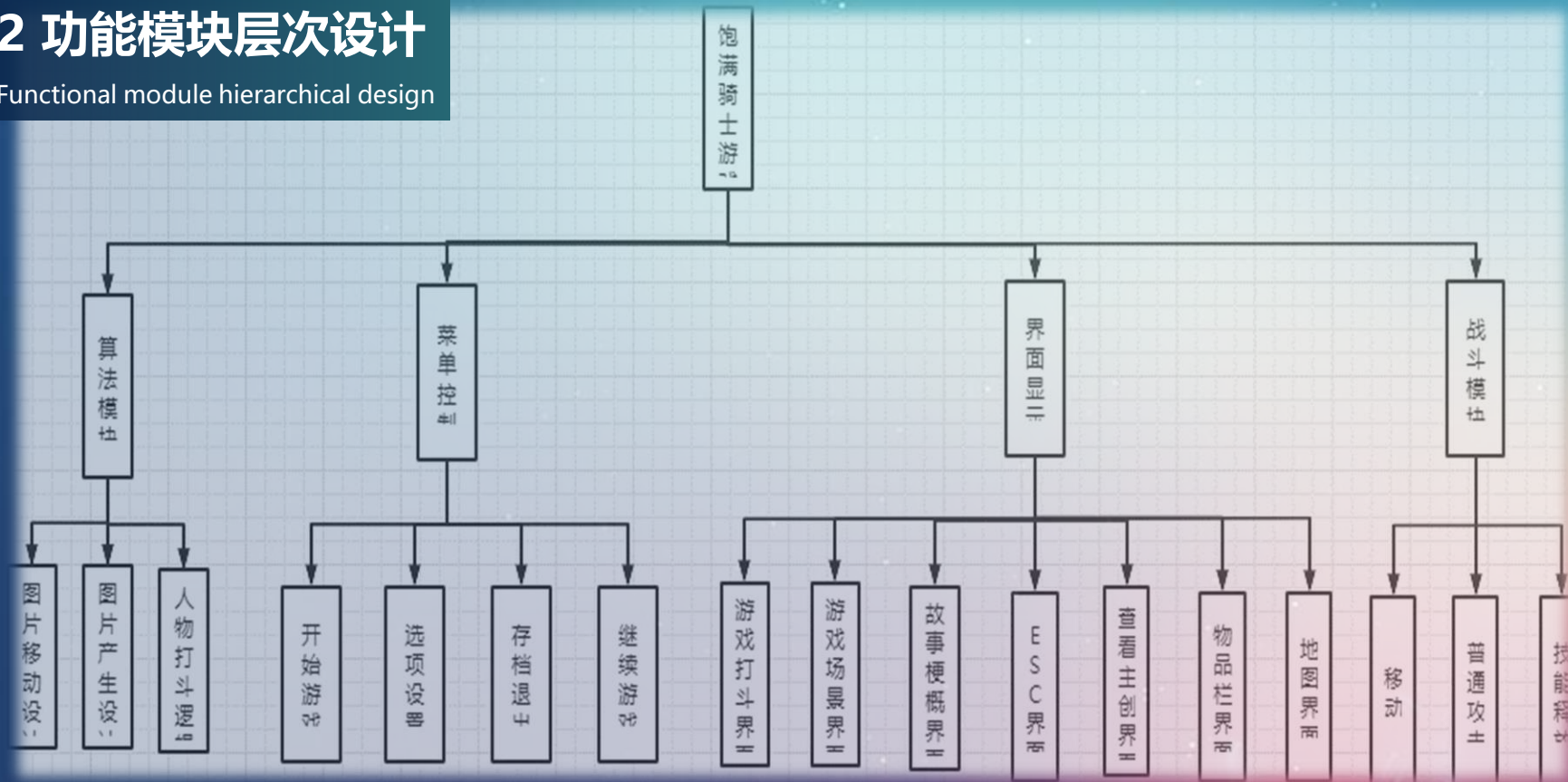
该层就是用来处理输入，比如当玩家走到怪物攻击范围内时，怪物会产生“做什么”的决策，例如攻击等，而这些“做什么”的决策，会封装到一个“请求”的结构中，然后向下一层传递。在该部分，我们使用了**Unity**的状态机，行为树和神经网络结构。玩家的输入也属于这一层，因为玩家的输入也是一种决策，这一部分就是通过行为树来进行处理，将玩家输入与**AI**可以共用

### 表现层

该层展示了游戏的**ui**界面，接受决策层的决策，显示相应**ui**。

## 3.2 功能模块层次设计

### 3.2 Functional module hierarchical design



功能模块层次图

Function module hierarchy diagram



## 3.2 功能模块层次设计

### 3.2 Functional module hierarchical design

#### 算法模块

- (1) 图片移动设计
- (2) 图片产生设计
- (3) 人物打斗逻辑

#### 界面显示模块

- (1) 游戏打斗界面
- (2) 游戏场景界面
- (3) 故事梗概界面
- (4) ESC界面
- (5) 查看主创界面
- (6) 物品栏界面
- (7) 地图界面

#### 菜单控制模块

- (1) 开始游戏
- (2) 选项设置
- (3) 存档退出
- (4) 继续游戏

#### 战斗模块

- (1) 移动
- (2) 普通攻击
- (3) 技能释放



## 3.2.1 算法模块

### 3.2.1 Algorithm module



#### 图片移动设计

在判断人物图片是否移动的逻辑上，如果玩家按了有效的移动按键（包括上下左右），则人物图片则可以相应地移动



#### 图片的产生设计

在玩家又一次攻克一个关卡以后，则地图则可以立刻标记那个关卡点，并在地图界面显示

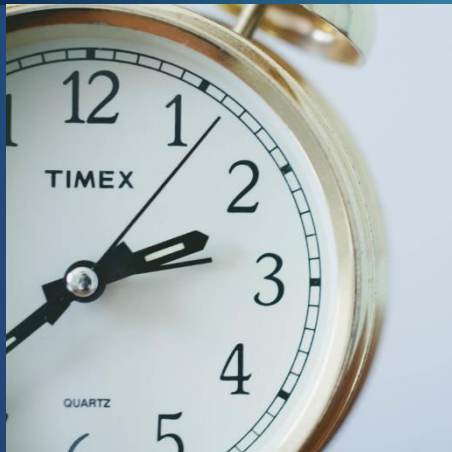


#### 人物打斗逻辑设计

其中怪物发射技能的时间间隔采用定时器算法，也就是指定具体的某一个时间值，怪物到指定的时间点后就可以开启攻击；而怪物采用技能攻击则是通过伪随机算法来实现，怪物是否开启技能攻击是随机发生的；怪物攻击的范围则是通过有限状态机做出响应。

## 3.2.2 菜单控制模块

### 3.2.2 Menu control module



#### 选项模块

进入“选项”界面可以设置音效、设置各个按键，包括向左、向右、向上、向下，和动作按键，包括跳跃、突进、抓住、交谈



#### 继续游戏

在“ESC”界面点击“继续游戏”则可以继续游戏

#### 开始游戏

进入游戏的主界面，有“开始”选项、“选项”选项、“主创”选项、“退出”选项，点击“开始”即开始游戏，进入存档界面



#### 存档退出

在“ESC”界面中点击“保存游戏并退出”则可以存档，下次在存档界面会有该存档记录。





## 3.2.3 界面显示模块

### 3.2.3 Interface display module

#### 1) 游戏打斗界面

人物与怪物打斗的场景，玩家可通过选项界面中的按键来攻击怪物，动作按键包括突进、抓住、跳跃、交谈

#### 2) 游戏场景界面

玩家可通过移动按键（包括向上向下向左向右）来在游戏场景中移动

#### 3) 故事梗概界面

玩家按“ESC”键进入ESC界面，点击选择“故事梗概”会出现原型设计界面中的“故事梗概”界面，待故事梗概呈现结束，玩家选择“下一步”进入游戏场景界面

#### 4) ESC界面

玩家点击“ESC”按键显示ESC界面

#### 5) 查看主创

点击主界面既可以查看主创人（界面暂时没做出来，期待后面成品）

#### 6) 物品栏界面

用户按下按键“B”可查看背包，出现原型设计界面中的“背包界面”

#### 7) 地图界面

玩家点击按键“M”进入原型设计界面的“地图界面”可查看地图，查看地图概貌和自己所处位置

## 3.2.4 战斗模块

### 3.2.4 Combat module

#### 1) 移动

玩家根据选项界面中的上下左右和跳跃按键来进行移动，其中上下左右和跳跃可以搭配使用

#### 2) 普通攻击

玩家可以通过按键“J”可以采用普通攻击

#### 3) 技能释放

玩家可以通过“u/i/o”按键来释放技能

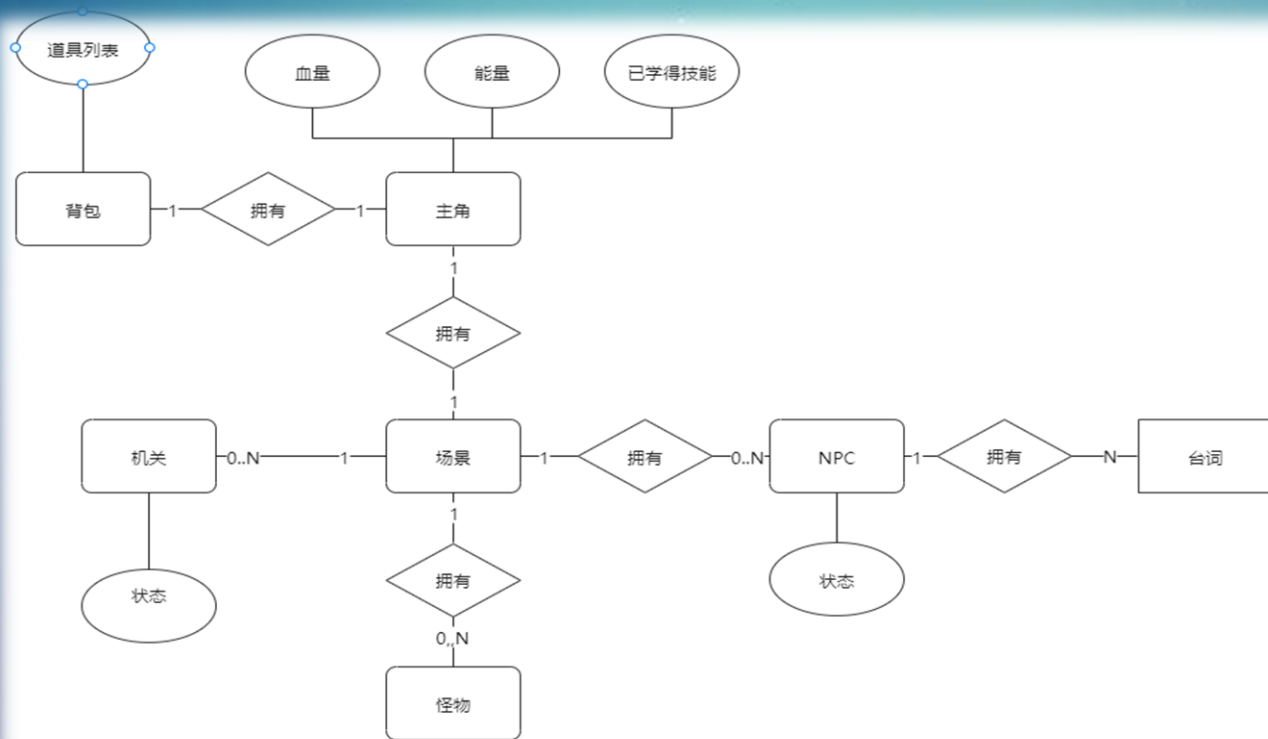
# 04.数据库设计

04. Database Design

---

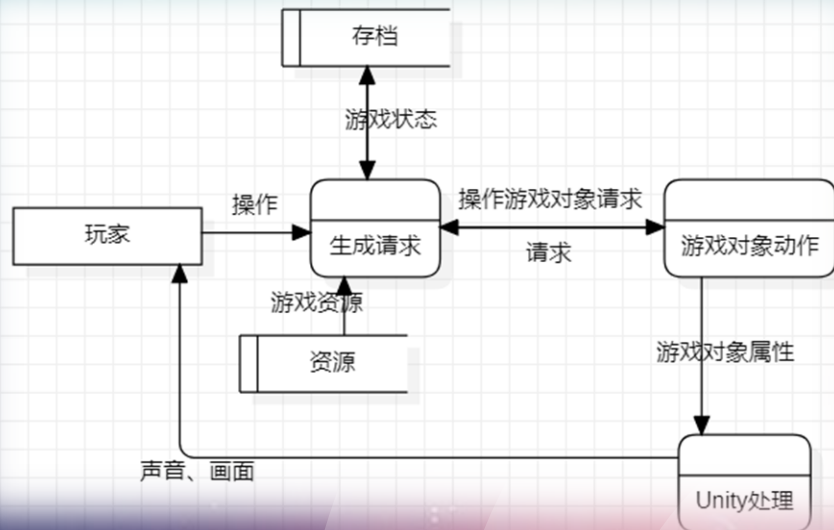
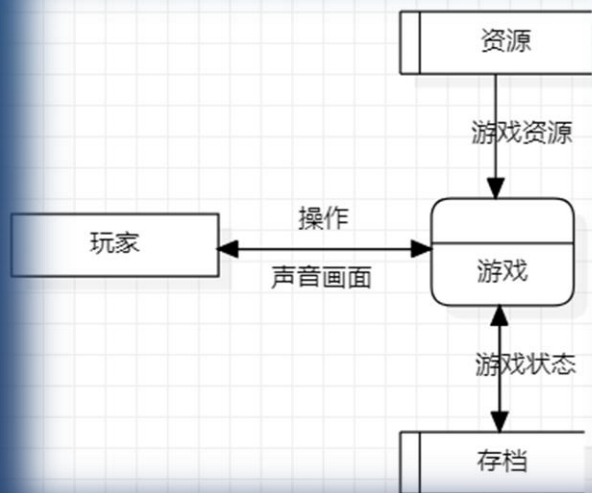
# 4.1 ER分析

## 4.1 ER analysis



# 4.1 数据流图

## 4.1 Data flow diagram



## 4.1 数据结构

### 4.1 Data Struct

静态数据保存描述信息，方便修改和多语言扩展

```
{ } item.json > { } achieveName
```

```
1 {  
2   "itemName" : { "name" : "", "decription" : "" },  
3   "achieveName" : { "name" : "", "decription" : "" }  
4 }
```

```
{ } map.json > ...
```

```
1 {  
2   "region" : "name"  
3 }
```

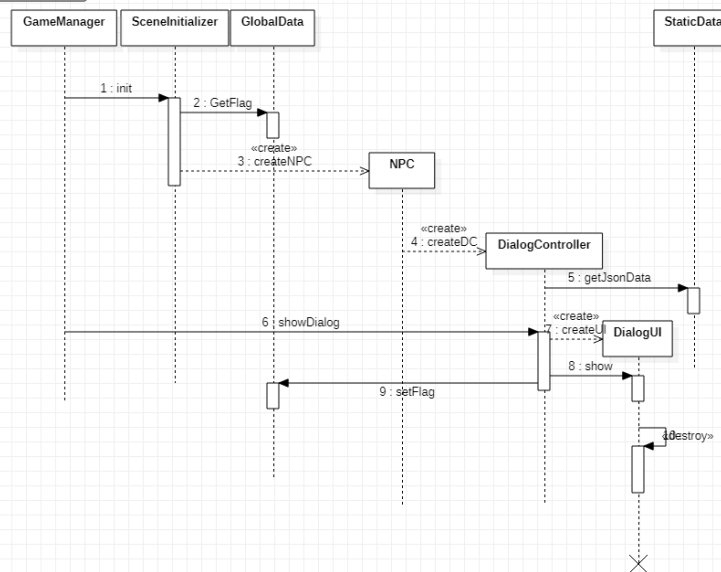
```
[  
  { "name" : "村民A",  
    "beforeMission" : [ "", "" ],  
    "onMission" : [ "", "" ],  
    "afterMission" : [ "", "" ],  
    "other" : [ "", "" ]  
  },  
  { "name" : "村民B",  
    "beforeMission" : [ "", "" ],  
    "afterMission" : [ "", "" ],  
    "other" : [ "", "" ]  
  }  
]
```

# 4.1 顺序图

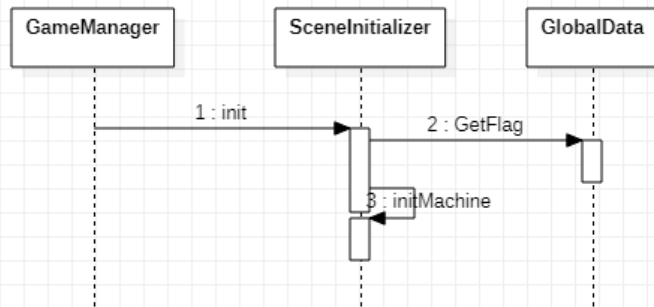
这些数据的使用方法

## 4.1 Sequence Diagram

sd SequenceDiagram1



sd 机关



# 05.类图与安全

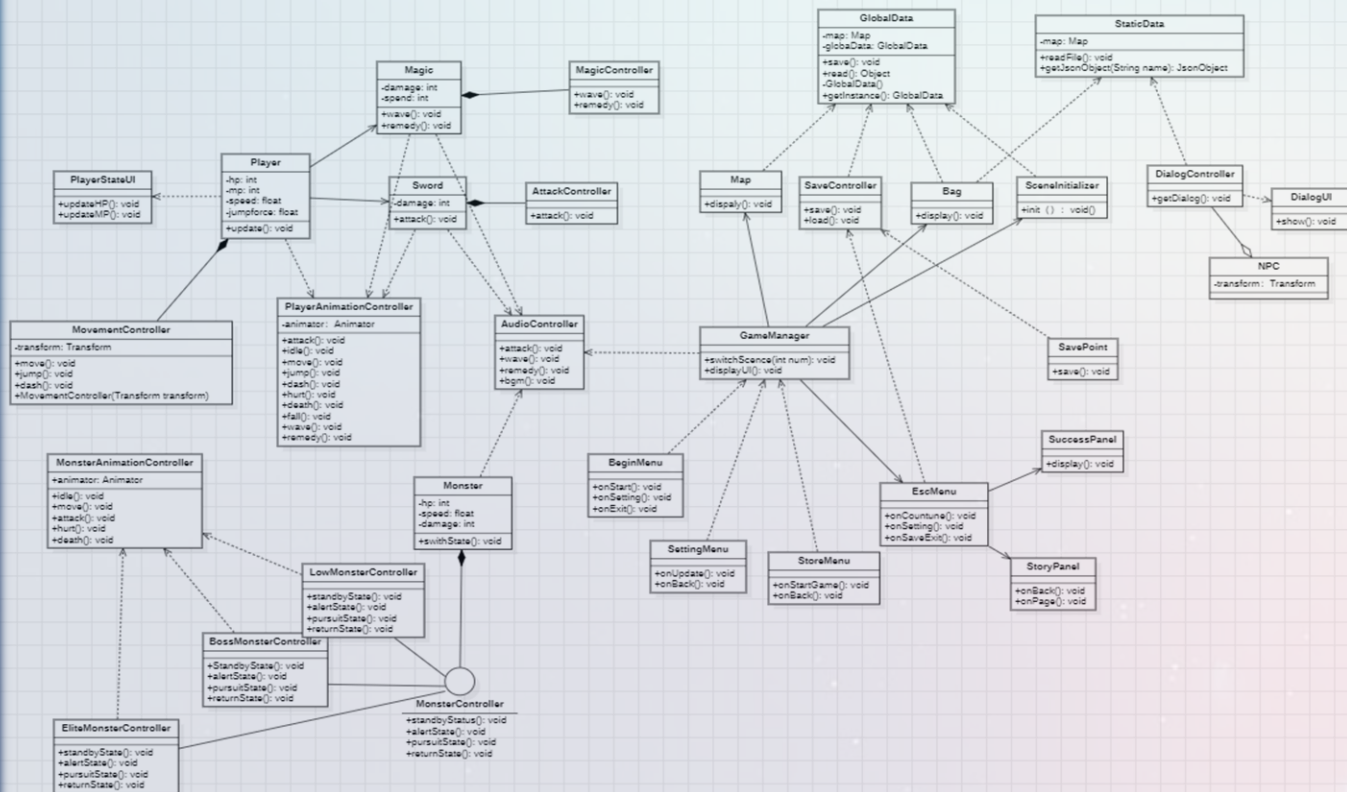
05. Class diagram and security

---



# 5.1 类图展示

## 5.1 Class diagram display



# 5.1 类图展示

## 5.1 Class diagram display

### 信息层

SceneInitializer

GlobalData

StaticData



### 决策层

BeginMenu

SettingMenu

StoreMenu

EscMenu

PlayerStateUI

DialogUI

StoryPanel

SuccessPanel

### 决策层

Player

Magic

Sword

GameManager

Map

Bag

Monster

NPC

SavePoint

### 行为层

MovementController

AttackController

MagicController

PlayerAnimationController

MonsterAnimationController

AudioController

SaveController

DialogController

## 5.2 安全设计

### 5.2 Safe design

使用 Unity 引擎开发游戏，易上手，效率高，成本低。Unity 引擎带来便捷和高效的同时，也引入了一些新的安全风险。游戏逆向分析的门槛也被大幅降低，分析人员的差距被拉平，即使没有太多逆向经验的人，也可以开始尝试制作外挂了。相比 C/C++ 游戏，Unity 游戏的安全风险存在分析工具多、破解门槛低、攻击方法通用化等几个特点。

Unity 支持 Mono 和 IL2CPP 两种编译模式。使用 Mono 模式编译的游戏，会将 C# 脚本代码编译为 IL 中间码，发布到游戏客户端。也就是源代码会泄露，这种中间码安全性较低，可以被一键反编译，一键修改。后来基于安全性和执行效率方面的考虑，Unity 支持了 IL2CPP 编译，大大提升了游戏安全性，但还是存在被攻击的风险。

因此我们团队决定使用 IL2CPP 模式编译我们的脚本代码，使用 IL2CPP 模式编译，游戏的脚本代码没有了，脚本代码被编译成了 Native 代码发布。很多反编译的工具都失效了，安全性得到了一定的提升。

## 5.3 权限设计

### 5.3 Permission design

1. 游戏的启动、暂停与退出权限
2. 主角操纵权限：玩家可以操纵游戏主角的移动、攻击和发动技能
3. 快捷键设置权限：玩家可以根据自己的习惯更改对主角操控的快捷键设置。
4. 背景音乐和音效设置权限：玩家可以更换背景音乐和音效
5. 查看地图权限：玩家可以查看地图了解自己的位置等信息
6. 对话权限：玩家在相应情景下可以和游戏对象进行对话交流以了解闯关相关信息。
7. 查看背包栏权限：玩家可以查看背包栏了解自己已经获得的物品。
8. 进入存档权限：玩家可以选择一个存在的存档并点击进入。
9. 创建存档权限：玩家可以创建一个新的游戏存档。
10. 删除存档权限：玩家可以删除一个存在的存档。
11. 故事梗概的查看权限：玩家可以打开故事梗概了解大概的故事。

# 绩效考核

Performance appraisal

学号	工作内容	贡献度
221801202	《数据的保存与恢复说明书》和数据的结构	11%
221801205	类图的设计及博客124部分	12%
221801211	系统体系结构的层次设计，ppt设计	12%
221801212	部分类图设计	8%
221801216	功能模块设计编写和博客第3567部分的编写	11%
221801228	数据库和ER图、数据的获取和传递、讲演报告	12%
221801229	系统安全和权限设计	11%
221801234	类图描述，《系统设计说明书》	13%
221801339	部分类图的设计	10%

# 2021

**感谢大家观看！**  
**Thanks for watching!**