编程规范

目的：本文档旨在规定和统一团队内部关于使用前端Unity的C#语言、后端Spring Boot的Java语言和数据库MySQL的编程规范。规范化编程风格有助于提升代码质量、增强代码可读性、简化维护过程并促进团队合作。

1. Unity C# 的编程规范

命名约定：

* 1. 类名：使用帕斯卡拼写法（PascalCase），所有单词的首字母为大写，单词之间没有连接符。例如‘RegisterButton’。
  2. 方法或函数名：使用PascalCase，例如‘CheckUserPassword’。
  3. 变量名：使用骆驼拼写法（CamelCase），例如‘bulletSpeed’。

代码结构：

1. 避免单个函数体过长，每个函数尽量不超过30行代码。
2. 重复代码应抽象成方法重用

注释：在复杂逻辑之前添加行注释，解释其目的和功能。

异常处理：使用Debug.Log处理可能发生异常的代码段，并对代码进行debug

1. Spring Boot Java 的编程规范

命名约定：

* 1. 类名：使用PascalCase，例如：‘GameDataServiceImpl’ 和 ‘GameDataController’
  2. 接口名和变量名：使用CamelCase，例如：‘getRankingS( )’ ，‘gameDataMapper’ 和 ‘gameData’

代码结构：

1. 按功能组织包结构，例如: ‘com.gameproject.gamebackend.controller;’,‘com.gameproject.gamebackend.service.impl;’和‘com.gameproject.gamebackend.entity;’。
2. 控制器层应只处理HTTP请求和响应，业务逻辑应在服务层处理。

注释：在每个类和公共方法上提供描述性注释，说明其功能和用途。在接口上添加注释，以方便实现前端和后端的接口连接。

1. MySQL编程规范

命名约定：表名使用小写字母，多个单词使用下划线连接，例如：‘user\_recordl’。

代码结构：使用适当的数据类型，避免过大的范围。为每张表设置主键。

注释：在表和重要字段上添加注释，说明其用途。