

MMPROG 数据库大作业

韩思绮 吴妍

一、 数据库设计

✧ player：存放用户基本信息

- dirty_scoin 存放用户离线中开始获得交易收益

	pname character varying (20)	password character varying (10)	p_id integer	health_point integer	level integer	s_coin integer	g_coin integer	dirty_scoin integer
1	Jekyll	3333	4	5	1	50	0	0
2	hsq	1111	1	5	2	63	79	0
3	qly	4444	5	5	2	54	30	0
4	wy	0000	2	5	1	6	90	0
5	Hyde	2222	3	5	1	230	3	0

✧ item：存放系统商场内固定出售的物品信息

	name character varying (10)	type character varying (20)	value integer	g_coin integer	s_coin integer	item_id [PK] integer	atype character varying (20)
1	Amulet_1	speed	11	5	10	1	accessory
2	Amulet_2	speed	15	6	12	2	accessory
3	Armor_1	defense	9	4	8	3	armour
4	Armor_2	defense	12	5	10	4	armour
5	Armor_3	defense	16	7	14	5	armour
6	Ax_1	attack	9	4	8	6	left_weapon
7	Ax_2	attack	12	6	12	7	left_weapon
8	Ax_3	attack	15	8	16	8	left_weapon
9	Bow	attack	13	7	14	9	right_weapon
10	Elixir_1	intelligence	9	5	10	10	medicine
11	Elixir_2	intelligence	7	3	6	11	medicine
12	Elixir_3	intelligence	8	4	8	12	medicine

✧ p_i：存放玩家物品拥有的对应关系

- 以 id 为 Key，每次插入时去当前 id 最大号+1 存储
- 若用户同时拥有多件物品，以单独的个体显示在背包中

	pname character varying (20)	iname character varying (10)	number integer	equipped integer	id integer
1	hsq	Elixir_5	1	0	43
2	wy	Shield	1	2	4
3	wy	Ax_3	1	2	5
4	hsq	Armor_1	1	0	26
5	wy	Amulet_1	1	0	16
6	wy	Amulet_2	1	0	29
7	wy	Ax_3	1	0	30
8	wy	Amulet_1	1	0	31
9	wy	Elixir_3	1	0	32
10	wy	Elixir_6	1	0	33
11	hsq	Elixir_6	1	0	34
12	wy	Helmet_1	1	0	35
13	wy	Elixir_2	1	0	36

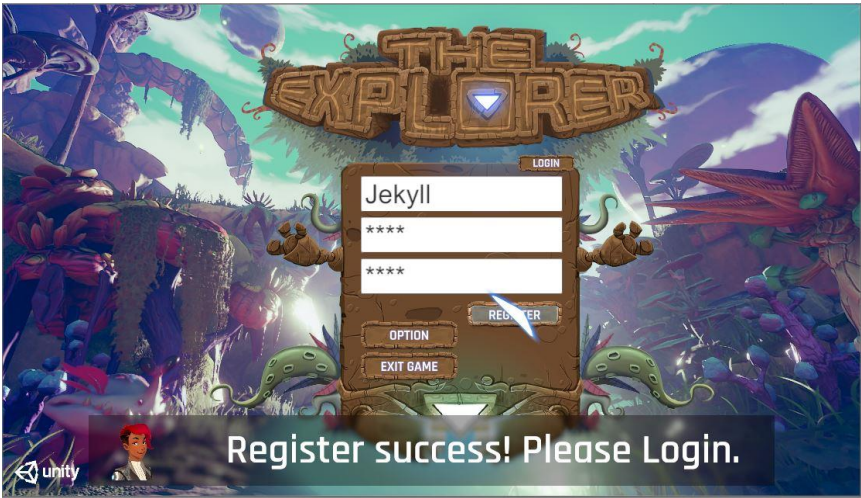
✧ trade：存放交易市场物品信息（出售者，p_i 中物品 iid 号，物品名，设定价格）

	uname character varying (20)	iid integer	iname character varying (10)	price integer
1	wy	5	Ax_3	8
2	qly	58	Elixir_6	10
3	qly	47	Sword_2	20

二、 功能实现及界面展示

✧ 登陆注册功能

✓ 注册

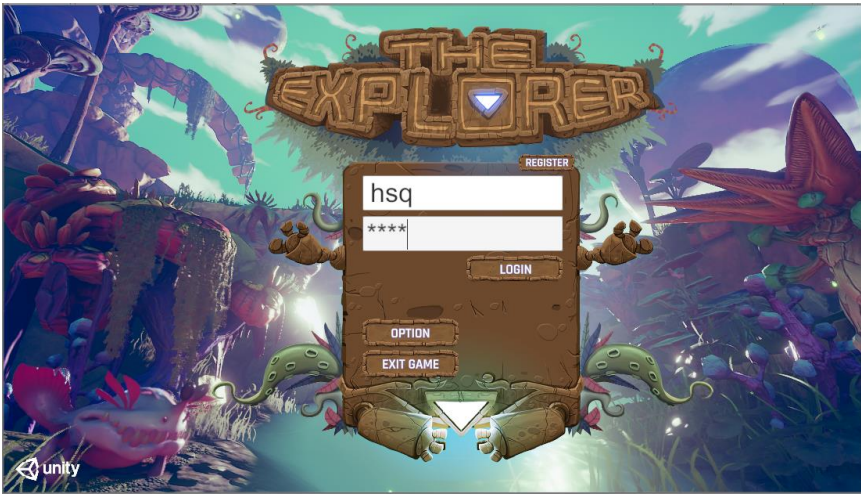


■ 后端

OnRecvRegister.cs

赋予新玩家 ID 和默认属性值，插入玩家表中

✓ 登陆



■ 后端

OnRecvLogin.cs

读取数据库，将玩家各个信息存入 player.cs

- ✓ 场景进入
 - 通信
 - 通过 SPlayerAttribute 信号传递
 - 前端
 - OnRecvPlayerAttribute.cs
 - 将所有信息存入前端全局类 PlayerAttribute 中

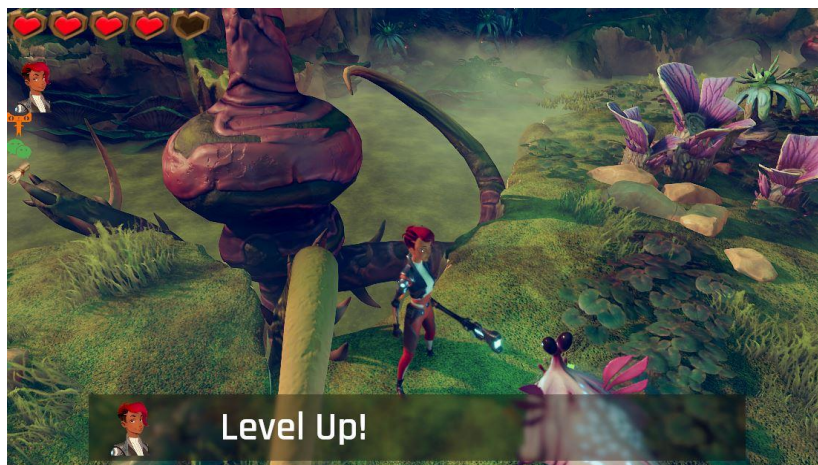
✧ 属性面板 RoleUI

- ✓ 属性值读取



- 前端
 - 读取全局类 PlayerAttribute 中信息，并在 RoleUI 中一一绑定显示。
 - 速度、智慧、防御、攻击属性值由佩戴宝物决定。
 - 生命属性值由玩家当前血量决定。

- ✓ 玩家升级
 - 玩家等级可由打败怪物方式升级。



✧ 背包功能

✓ 背包物品属性

■ 后端

SerializableData.cs 序列化创建 M_Item 类用于存放物品的各项属性以及两种枚举类型。

一个物品对应一种类型及一种属性加成。

■ 前端

读取全局类 PlayerAttribute 中玩家背包信息（以 Dictionary 形式），并在 InventoryUI 中绑定显示。

✓ 背包物品佩戴及卸下



■ 前端

InventoryUI 中绑定 button 脚本

■ 通信（前->后）

通过 CItemApply 信号传递

■ 后端

OnRecvItemApply.cs

将信息和 player.cs 中玩家宝物佩戴情况进行对比整合，判断操作是否成功，若成功修改数据库，发送成功信号。

*可以区分一类物品、一种物品

■ 通信（后->前）

通过 SItemApply 信号传递

■ 前端

OnRecvApplySuccess.cs

操作成功则修改对应 PlayerAttribute 中物品相关属性，并刷新界面信息（属性值、佩戴情况、佩戴数量）

✓ 背包物品挂牌及撤销挂牌



✓ 背包已佩戴物品界面显示

EquippedUI.cs



- ✓ 背包待售物品界面显示



✧ 商场功能

- ✓ 前端界面设计

- 增加商品详情及拥有情况 ShelfItemUI.cs
创建 DetailsView 弹窗视图，从 PlayerAttribute 中读取显示商品详细信息
- 增加金币银币余额数显示
- 增加金币银币当前购物车总额显示
- 金币银币购买按钮脚本 CartGridUI.cs



- ✓ 购买逻辑实现

- 通信（前->后）
通过 CBuyItems 信号传递
- 后端
OnRecvBuyItems.cs
确认钱币足够并更新数据库及 player.cs 中背包情况，发送成功信号
*更新数据库时对于 ID 号设置，可以唯一标识用户及该项物品
- 通信（后->前）
通过 SItemBuy 信号传递

- 前端
OnRecvItemBuy.cs
PlayerAttribute 背包中加入新物品，并更新金币数，刷新界面信息

✧ 金币银币充值兑换功能

✓ 金币充值

- 前端
增加 RechargeGoldUI.cs
放置二维码及充值数额填入框，假定用户充值完毕，向后端发送信号并刷新用户的金币数
- 通信（前->后）
通过 CAddCoin 信号传递
- 后端
OnRecvAddCoins.cs
更新数据库和用户金币数



✓ 银币兑换

- 前端
增加 RechargeSilverUI.cs
假定金币兑换银币比率为 1:2，放置数额填入框。确认兑换后，向后端发送信号并刷新用户的金币数和银币数。
- 通信（前->后）
通过 CAddCoin 信号传递
- 后端
OnRecvAddCoins.cs
更新数据库和用户金币数



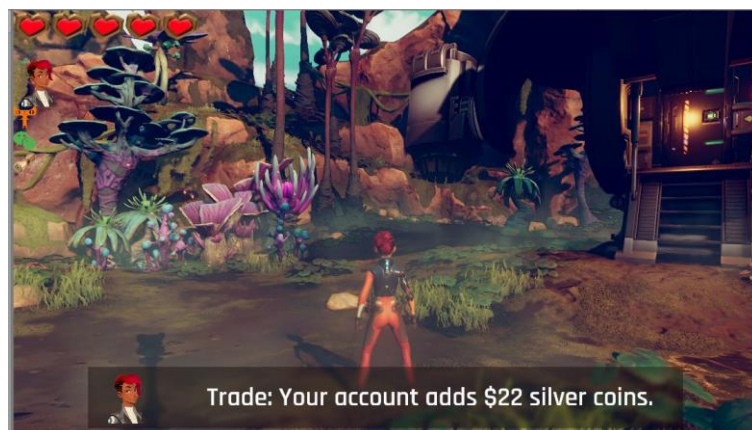
✧ 交易商店功能

✓ 前端界面设计（以商场初始界面为基础）

- 增加商品详情及拥有情况 TradeShelfItemUI.cs
 - 创建 TDetailsView 弹窗视图
 - 从 Trade_item 中读取显示商品详细信息及所有者，需新建字典类型
- 增加金币银币余额数显示
- 增加银币当前购物车总额显示
- 银币购买按钮脚本 TradeCartGridUI.cs
- 撤销商品数量增加减少功能，重新设置移出购物车的方法

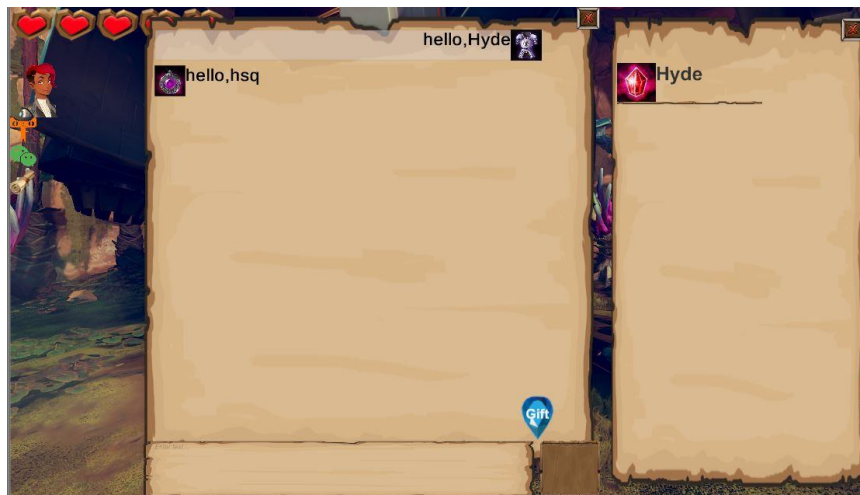


出售方获得交易收益提示：



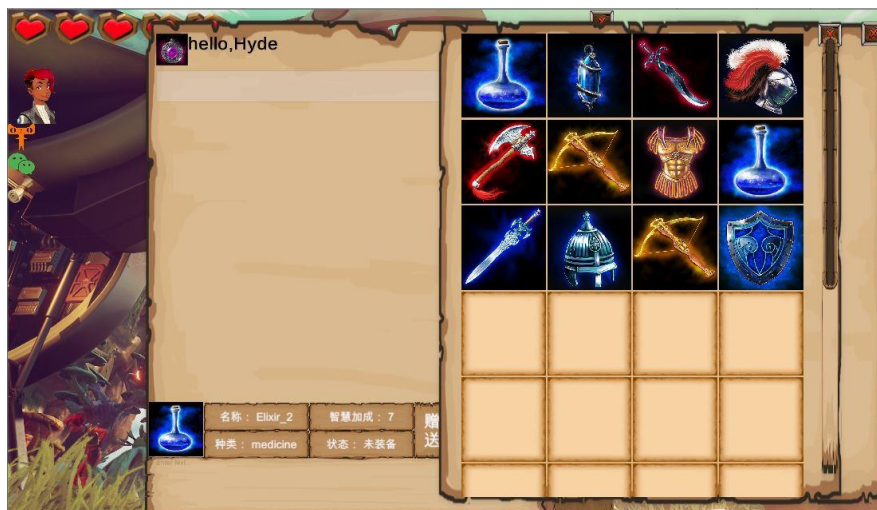
✧ 聊天系统功能

- ✓ 前端界面(使用默认)
- ✓ 前后端通信
 - 提取当前场景中所有玩家（除用户自身）作为可聊天对象。
FriendUI.cs OnRecvGetFriends.cs
 - 点击图标传递好友姓名 FriendInfoUI.cs ChatMessage.cs
 - 用户打开和聊天对象的对话框可以接收到自该用户上线后接受到的和该聊天对象的所有对话内容。（由后端发送消息获得） Chat.cs
 - 向聊天对象发送消息 OnRecvSendMessage.cs
 - 接收来自聊天对象的消息 OnRecvRecvMessage.cs

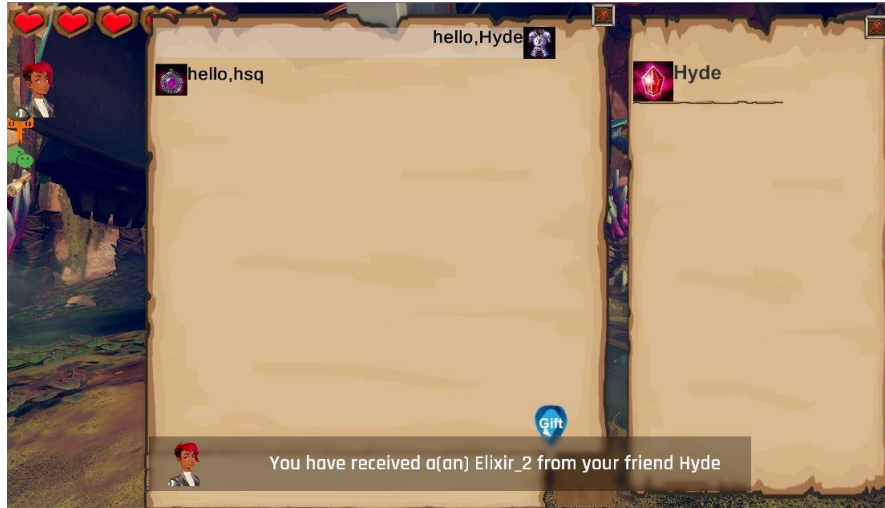


✧ 好友赠礼功能

- 可以通过对话框中的按钮打开背包选择礼物赠送给朋友。GiftUI.cs



收礼方获得礼品提示：



三、 总结

历时一个多月的大作业在期末考前一周顺利验收,我们两个人也在此次的期末大作业展示中收获了许多。尤其是数据库方面,从表的设计到 SQL 语句的应用,都得到了许多操练,并且对一些理论知识也进行了巩固和理解的提升。

大家精彩的展示也让我们意识到了做到现在这个程度还是远远有提升空间的,在完善和开发新功能方面我们得到了许多有意义的启发。比如在游戏的数据库性能优化上,可以用到 cache 定时刷盘的机制来减少数据库操作的耗时;比如在容错方面我们还可以进一步进行备份和回滚机制,有了新的认识和考量;在游戏玩法上也有许多新的创意值得我们学习。

在这次的 Unity 3D 开发中我们虽然遇到了很多困难,但是在解决问题的过程中我们也在不断进步,在实践中得到了通过一般的理论学习所得不到的宝贵体验。我们会珍惜这次难得的 Unity 3D 开发经验,并将收获到的知识与工作技能运用到之后的学习中去。

四、 小组分工

韩思绮:数据库设计、登录注册、属性面板、背包相关功能、物品佩戴实现、商场购物、兑换充值功能、交易前端

吴妍:数据库设计、好友聊天、赠送礼物功能、玩家升级、交易逻辑实现、物品挂牌实现