



IOS平台网络游戏服务端的设计与实现

李自勉





背景介绍

卡牌游戏是一种种类繁多、 受众面广的游戏类型, 也是 桌面游戏的一种。卡牌游戏 主要分为两类, 即集换式卡 牌游戏与非集换式卡牌游戏。 非集换式卡牌游戏包括常见 的扑克牌和现在非常流行的 三国杀等, 集换式卡牌游戏 则包括万智牌、游戏王等。



背景介绍

- IOS平台卡牌游戏的优势:
- 1.集换式卡牌游戏具有收集要素,这一点可以提高玩家的沉迷度,并且创造稳定的盈利点;
- ② 2.在虚拟的平台上集换式卡牌游戏的卡牌间的互动规则设计的开放程度更大,甚至可以抛开卡牌的要素去设计交互规则;
- 3.集换式卡牌游戏玩家间除去基于游戏规则的对抗或者合作,还有交易要素,这一点也可以作为一个盈利点。



项目目的

- 学习使用Go语言进行网络并行程序开发;
- 实现一个能够持续正常运行的服务端,服务端可以不断接收新的连接,完成用户创建与登录工作。玩家进行对战要建立玩家的排队队列,然后分配每一局游戏的玩家与对手,每局对战过程中的逻辑运算要由服务端计算完成,并且把计算结果用消息发送给客户端,并接受客户端的反馈。



开发工具和环境

● 开发语言: Go语言1.1

● 开发系统: MacOS 10.8

数据库软件: MySQL 5.5.16 for Mac

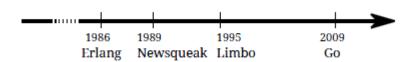


相关技术介绍

Go语言



Go语言是Google开发的一种编译型,并发型,并具有垃圾回收功能的编程语言。于2009年11月由正式推出。

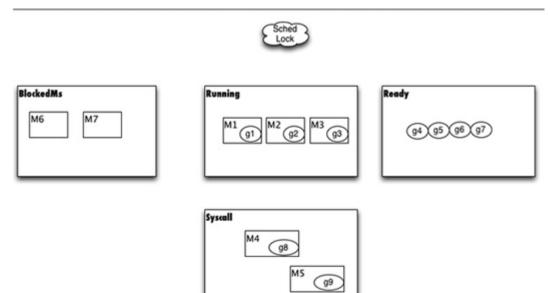




相关技术介绍

Go语言——Goroutine

Goroutine是Go语言在语言级别支持并发的一种特性,Go语言最大的特点就是线程轻量级的实现,goroutine配合channel形成了Go语言处理并发的基础。但goroutine的实现并不是真正的并发。





相关技术介绍

Go语言——interface

在Go 中的接口有着与许多其他编程语言类似的思路: C++ 中的纯抽象虚基类,Haskell 中的typeclasses 或者Python 中的duck typing。然而没有其他任何一 个语言联合了接口值、静态类型检查、运行时动态 转换,以及无须明确定义类型适配一个接口。







文件目录结构

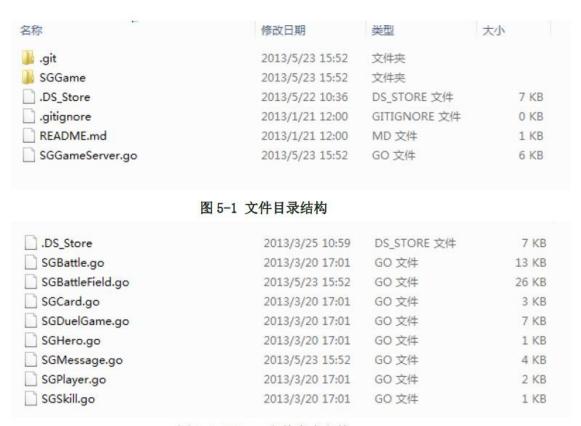
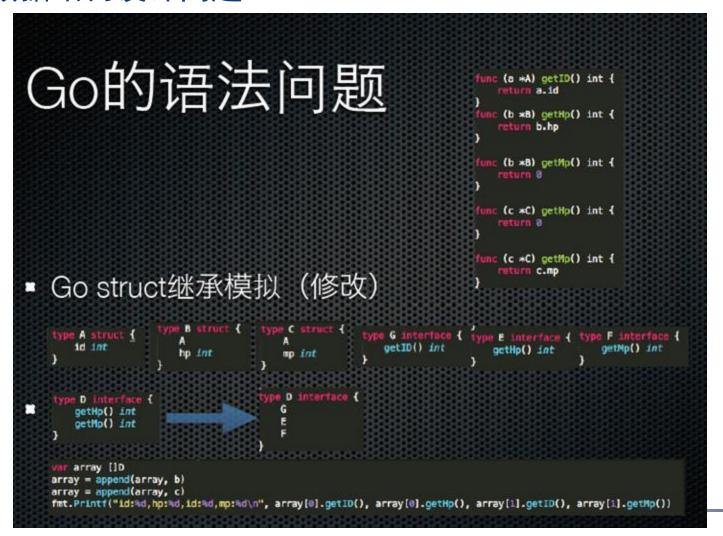


图 5-2 SGGame 文件夹内文件



- ◉ 服务端构建:
- Goroutine1: 监听与保存外部连接
- Goroutine2: 发送心跳, 维持连接
- Goroutine3: 对排队玩家进行匹配
- Goroutine4: 游戏对局

● 数据结构设计问题



```
消息设计
消息类的结构:
type DataInfo struct {
infoType int32
}
```

- 通信包的结构:(前4位)长度+(1位)信息类型+信息内容
- 消息封装: JSON



- ◉ 游戏规则简要说明:
- 每位玩家具有初始给定的血量和一副套牌,套牌中暂定由士兵卡牌和技能卡牌构成;
- 游戏按照每个玩家一回合,轮回执行;
- 当有任意一位玩家血量降为0时,则该玩家该局游戏失败 ,对方玩家胜利。

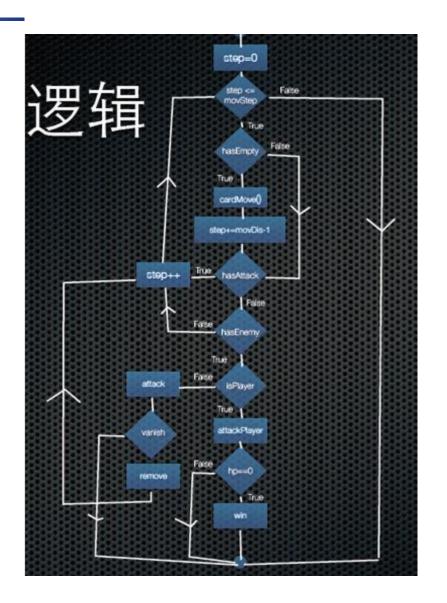


- 游戏规则简要说明:
- 每个玩家回合分为4个阶段,分别为抽牌阶段、放置阶段、 战斗阶段和结束阶段:
- 抽牌阶段,无特殊情况玩家抽取1张卡牌。在游戏刚开始时双方玩家都要先抽取2张卡牌才进行后面的回合。
- 放置阶段,玩家将可以使用的士兵卡牌放置在战场上, 或者使用技能卡牌。
- 战斗阶段,玩家放置好的卡牌进行行动,包括移动和攻击以及卡牌技能的发动与使用。
- 结束阶段,更新玩家手牌的状态,并检查手牌数是否超过上限,超过则弃掉超过数目的牌。



● 士兵逻辑设计:

士兵牌可以简化分为2种类型:近战士兵和远程士兵。近战士兵是靠近敌人后攻击,远程士兵是在最远距离攻击敌人。





● 技能设计:

1	SKILLID	名字	效果
2	S1001	奔跑	移动+1
3	S1002	骑兵	移动+2
4	S1003	钝步	移动-1
5	S1004	锁足	移动-2
6	S1005	短弓	射程+2
7	S1006	弓箭	射程+3
8	S1007	长弓	射程+4
9	S1008	飞行	移动时可穿越敌方单位
10	S1009	铁甲1	减免受到的物理伤害1点
11	S1010	铁甲2	减免受到的物理伤害2点
12	S1011	铁甲3	减免受到的物理伤害3点
13	S1012	法抗1	减免受到的法术伤害1点
14	S1013	法抗2	减免受到的法术伤害2点
15	S1014	法抗3	减免受到的法术伤害3点
16	S1015	治疗1	每回合为受到最多伤害的友方士兵回复1点生命
17	S1016	治疗2	每回合为受到最多伤害的友方士兵回复2点生命
18	S1017	治疗3	每回合为受到最多伤害的友方士兵回复3点生命
19	S1018	治疗4	每回合为受到最多伤害的友方士兵回复4点生命
20	S1019	吸血	造成伤害时,对自己治疗等量的伤害
21	S1020	反击	受到射程内的攻击时,反击攻击者
22	S1021	破甲	造成的伤害无法被减免
23	S1022	警戒	优先攻击身后的单位



◉ 调试输出:

```
Game Message : Turn 1
Game Message: Player tony Action, Hp:20
Game Message : Turn 2
Game Message: Player lizimian Action, Hp:20
Testing Message: Place Card Information: {"CardIndex": 0. "Position": 108}
Testing Message: Card index 0, place to (0,8)
Testing Message: Card (0.8) move to (0.5)
Game Message : Turn 3
Game Message : Player tony Action, Hp:20
Testing Message: Place Card Information: {"CardIndex": 3, "Position": 101}
Testing Message: Card index 3, place to (0.1)
Testing Message: Place Card Information: {"CardIndex": 2, "Position": 201}
Testing Message: Card index 2, place to (1,1)
Testing Message: Place Card Information: {"CardIndex": 1, "Position": 200}
Testing Message: Card index 1, place to (1,0)
Testing Message: Place Card Information: {"CardIndex": 0, "Position": 190}
Testing Message: Card index 0, place to (0.0)
Testing Message: Card (0,1) move to (0,4)
Testing Message: Card (0,4) attack to (0,5)
Testing Message: Card (0.5) removed
Testing Message: Card (0,0) nove to (0,2)
Testing Message: Card (1,1) nove to (1,3)
Testing Message: Card (1,0) move to (1,2)
```



- 本次项目基本完成了最初的设计要求,在后面的时间里仍然有机会继续扩展与改进。
- 在项目的编写过程中,我对Go语言有了一个初步的了解,对服务端架构的设计和通讯协议的设计也有了了解,这都是非常重要的经验。





Q&A







谢谢!

