Go关于总线式通信的想法

## 想法来源

经过Gorouting 的学习以及近期关于游戏开发的简单了解，以及RPC通讯的了解，产生的简单想法。

## RPC 通讯

通过leaf的学习，以及网上的资料，简单可以理解为远程函数调用。即客户端或服务器通过消息通讯传送参数调用其他服务器的函数。比如leaf 登录网页发送login 数据到leaf服务器，调用login 模块的handleTest 函数。这种调用目前看来是一对一的，如果是对战游戏，简单场景的交互可以容易实现，如果是复杂场景，游戏人物和游戏场景的互动极有可能会变得很复杂。【没开发过游戏，只是简单猜想，如果把游戏场景不当做背景，都当做和其他角色一样的可交互对象，则一个角色要同事和多个交互物体通信，而其他物体也同时和其他多个物体通讯】 则一对一的通讯会显得低效，且逻辑复杂，需要一种有效的多对多通信方式。

## 定义总线式通讯。

简单的说类似电脑主板总线的概念，是一个通用的通讯链路，实现多对多的通讯方案。对于Go语言，是一个多发送 多接受的channel

1. 模型设计

总线1

对象3

对象2

对象1

对象4

1. 概念定义

* 对象

可以对消息做出反馈的单个虚拟实例。比如游戏中玩家控制的角色，npc等。如果如果没有反馈则不属于对象，比如一些游戏中的树，无法交互，不属于对象。

每个对象都有自己的独立id

* 总线

是一类对象的消息通讯中心，即消息的集合，每一类对象应该有自己单独的消息总线。

每一个具体对象可以属于多个不同的总线。

对象所属的不同总线代表了对象所支持的不同属性的消息

相同属性的总线可以存在多个，每个总线所包含的对象数量可以不同。

同一个对象可以在一定条件下脱离一个总线或加入一个总线

* 消息

对象之间通过总线发送消息进行通讯

当一个对象发送消息时，总线上其他所有的对象都会收到消息

消息的返回是一个新的消息

* 空消息

总线不处理空消息

* 独立消息

单独发送给特定对象的消息，只能发送一个特定对象

1. 总线内部消息的转发处理

* 对象和总线的通讯通过channel实现，当对象向总线的channel写入消息时，触发转发，向总线内其他对象的channel写入消息
* 由于对象总是对消息做出反馈，避免消息风暴，可以反馈空消息，但设计时尽量避免空消息

1. 对象的内部处理设计

* 消息处理

对象每一次接受到消息都是其他对象和自身的交互，比如伤害，移动，任务提交完成等等。

每次消息处理必须有一个新的消息作为反馈发送出去，但新的消息可以发送到其他的总线

* 加入\离开\新建总线

对象可以自己判定满足一定条件加入加入\离开\新建总线，也可以根据一定条件将其他对象加入自己所存在的总线。

## 设计目标

1. 实现多对多的对象通讯交互

总线设计模型类似局域网的单播和多播模式，而对象既是消息的处理方也是总线之间通讯的路由

1. 实现不同对象之间的非接触式通讯

如果我之发出一条消息，而不关心具体的消息接受对象，而消息传播过程中的演化也不关心，比如游戏中点燃一片草地，草地燃烧扩散，扩散过程中消融冰系魔法，造成伤害等等

## 远期目标

1. 通过总线网络构建消息网络

个对象可以连接多个总线，所以对象是消息的节点，而一群对象包含多组总线就构成了消息网络。

在这个网络中，每一个对象都可以看做一个神经元，不同种类的对象就是不同种类的神经元细胞，总线不负责消息的逻辑处理，整个网络可以构建成神经元网络

1. 通过机器学习实现消息内部逻辑处理。

对同一类的对象，可以共用一个消息逻辑处理引擎，而消息的处理可以抽象成一个输入一个输出，输入和输出都有多个不同的枚举类型，输出的值由输入的值确定，可以考虑通过机器学习实现。

1. AI

神经元网络有了，每个节点都是机器学习的，如果再实现消息类型的自我新增和删除，对象类型的自我新增和删除，只要网络规模足够大，可以通过图灵测试的AI就实现了