2D类仙剑1游戏

# 游戏开发背景

为了中国游戏的未来，为了不让欧美鬼畜游戏主宰中国游戏界，我和玉子毅然决然的投身游戏开发行业。好吧，说出来我也不信。真相是某一天玉子说我们来开发游戏，我说好，于是我们就开始着手准备这玩意。

实际上一开始我们也不知道该做个啥，3D游戏嘛，逼格是挺高的但是开发难度比喜马拉雅山还高，肯定是做不了的，做个弹幕游戏又没啥意思。后来想了想其实我们可以做个类似仙剑1的2D游戏。好吧，说干就干。

# 游戏流程简述

游戏玩家启动游戏程序后重新开始新游戏或者选取以前的存档。

游戏开始后进入游戏场景，在场景中遇到怪物或者NPC，如果是怪物则进入打怪流程，完成打怪后得到经验，或者死了需重新开始，或者中途逃走；如果是NPC则开始NPC对话场景，完成对话之后，或者得到线索，或者没有任何收获只是简单对话；特殊场景中还能获取宝箱，宝箱中有装备、药物、金钱等。完成一定剧情之后可以出发剧情对话，剧情对话中可以获得奖励，或者什么也没有。

# 游戏流程算法

当进入场景后，最初人物并未移动，静态物，如房屋、树木、草地、山岭等不会移动，NPC会在设定范围内随机运动。

当鼠标点击场景时，获取鼠标位置，系统计算鼠标位置是否落于人物可行走范围外，如果是，则计算人物于鼠标落点两点之间可行走到的边界点，人物行走至此点，否则人物直接行走至鼠标落点。

当人物行走时，系统计算人物于最近点怪物的距离，如果该距离小于或等于设定的怪物追逐范围，怪物将追逐人物直至两者相遇进入打怪流程，或者人物逃避，系统计算追逐怪物与人物之间的距离是否超过设定的追逐范围，如果是怪物返回原来的位置。

# 游戏素材

游戏中的人物、npc、spirit、地图、树木、房屋等等在游戏中的可视部分都是图片。

# 游戏界面

游戏界面有一份界面设计文档和使用界面Axure设计的界面工程。在界面设计完毕后可以在UML中编制界面元素类型及其相关关系。

关卡界面由三个层次对象组成。背景地图为第三层、人物npc怪物为第二层、树木等为第一层。第一层遮挡第二层、第二层遮挡第三层。并且对于关卡界面，从上往下一次遮挡对象的重叠部分。比如npc1在上面，npc2在下面，npc2的头和npc1的脚重叠，则npc1的脚遮挡npc2的头，关卡数据结构需要考虑这个。

# 游戏资源管理

目前暂定如下几个模块，内存管理模块，对象管理模块，场景管理模块、数学模块、物理模块、碰撞检测模块、脚本解析模块。

内存管理模块管理游戏中所有对象内存的分配。

对象管理模块管理游戏中所有对象的生成和消除。

场景管理模块管理游戏中所有场景中的静态和动态对象。

数学模块管理游戏中数学部分，包括形状，比如圆形、矩形、线、点、向量、矩阵等，以及数学计算，比如距离、尺寸、数值乘除法、向量及矩阵乘除法等。

物理模块管理游戏中与物理相关方面，比如力学的计算。

碰撞检测模块管理游戏中物体碰撞，比如边界框、边界球、平面碰撞等。

脚本解析模块管理游戏中场景脚本的解析。

数据库管理模块管理游戏中人物NPC怪物等的部分属性。

任务管理模块管理游戏中的任务。

剧情管理模块管理游戏中的剧情。剧情属于隐藏任务，并非是从npc那里获取的任务，而是完成某个剧情之后才会在指定npc处触发。

# 辅助工具

数据库管理工具，辅助添加、删除、修改，数据库中的内容。

地图编辑工具，编辑地图。

# 游戏启动

游戏启动后显示启动界面，提供新的开始和读取存档选择。

1. 新的开始

脚本模块读取初始脚本文件，脚本文件代表一个关卡，建立脚本对象也就是关卡对象。

场景管理模块根据脚本对象中的元素向对象管理模块生成对象加入场景对象组中。

对象管理模块向内存管理模块申请内存建立对象。

1. 读取存档

如果是读取存档，则需要将当时的场景及人物序列化输出为存档文件。

# 数据结构与算法

1. 脚本模块

* 语法对象

脚本语法符号定义在脚本语法文件中。在脚本模块初始化的时候建立脚本语法对象，读取脚本语法文件，将脚本符号与代表该符号的整形值存储在树中，语法树是二叉树。目前以链表形式表示。在脚本语法中进行定义是为了方便游戏开发者可以通过定义语法的同名词进行定义。语法文件中语法与符号关系如下，grammer:ID。;英文分号是注释。

* 场景脚本对象

游戏可以直接使用C++进行编程，也可以使用脚本文件设置场景。脚本场景文件中区分静态对象和动态对象，静态对象是背景，动态对象是可以与之交互的怪物、NPC、宝箱等。脚本场景文件中表示的每个游戏对象都需要一些共同的属性，位置、大小、图片路径、类型(静态、动态)、如果是动态还有攻击力、活动范围、血量、活动范围、追击范围等。

一个关卡对象代表一个脚本文件，脚本模块读取脚本文件的时候建立脚本对象。脚本文件中每一个对象由一个关卡元素对象代表。

关卡对象有三种类型，静态不能交互的对象，动态可以交互的对象，上一个或下一个脚本的链接对象。静态对象和动态对象，有一个数据库中该对象类型的ID、场景中的对象id、关卡对象名字、名字、数量、是否有位置标识、位置、普通对话文件路径、任务对话文件路径、剧情对话文件路径。每个关卡对象由链表连接起来。

因为XML文件中可能某种怪物有很多个，不可能每个相同的怪物都设置名字、位置等，而且他们都有相同的名字和图片，因此对象结构定义一个数量、是否有位置标识，如果没有位置标识，则随机生成一个地图中可达的位置。

因此脚本模块中有一个关卡元素结构，一个关卡对象结构。

脚本文件读取文件，建立关卡对象，添加关卡对象到脚本对象中。

具体结构内容详见UML。

1. 场景管理模块

场景管理模块维护一个游戏对象链表，根据脚本模块返回的对象类型在数据数据库中的ID向对象管理模块申请对象，场景管理模块将该对象加入游戏对象链表中。

当进入下一个场景后，将当前场景返回到对象管理模块中。

1. 对象管理模块

对象管理模块维护一个游戏对象链表。当场景管理模块向对象管理模块申请对象的时候，对象管理模块检查游戏对象链表中是否有相同数据库ID的对象，如果有则返回该对象指针，如果没有则向内存管理器申请该对象大小的对象句柄。同时利用对象句柄函数类型设置内存类型。

对象句柄类型设置函数会利用对象管理模块传进来的对象类型调用对象生成器生成对象，对象生成器会有一个巨大的switch语句根据对象类型将对象句柄中的内存转换为相应的对象类型，利用数据库对象ID(也就是对象类型)向数据库管理模块读取相应对象属性，并设置相应属性。当场景管理模块返回场景到对象管理模块中时，对象管理模块先不将对象返回到内存管理模块中，而是将对象存储在对象管理模块的对象链表中。在某一个时刻在将没有使用到的内存进行释放。

1. 内存管理模块

内存管理模块管理游戏中所有对象内存的分配。

目前我想的方法是内存管理模块初始化时分配一块大内存，具体有多大可以自行设置或者进行配置(在初始化配置文件文件中)。分配内存的时候需要考虑到对齐，可以查看游戏编程精粹1中的1.9部分。

对于每块分配的内存，内存管理模块都要建立一个内存句柄，存储分配的内存块的起始内存地址指针。内存管理模块在调整内存块的位置，消除整个内存块的内存碎片时，需要更新对象句柄中的指针。内存管理模块消除内存块中的内存碎片的时机需要小心确定。

使用内存的其他模块对于每个对象都有一个对象句柄，对象句柄中存储内存句柄的指针，管理所有分配的内存。具体结构查看UML图。每个使用内存的模块，根据所需大小向内存管理模块申请内存句柄，并将分配的内存句柄加入到对象句柄中。当使用对象句柄获取对象内容时需要将内存句柄中的内存指针用static\_cast转换为所需要的对象指针类型。

1. 数据库管理模块

数据库使用sqlite3，存储人物npc怪物的血量、攻击力、活动半径、攻击半径、攻击方式比如武技仙术这些。

建立两张表格。

第一张表格，存储人物、npc、怪物的ID、血量、攻击力、活动半径、攻击半径、独特武技。以ID或者名字作为主键，第二张表格是关联表。

第二张表格，存储人物、npc、怪物的ID或者名字攻击方式，仙术(仙术名字、伤害、图片或者图片路径)、学习等级等。