**手机安卓端app详细设计说明书**

****

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [ ] 正式发布  [√] 正在修改 | 文件标识： | SE2018春-G08-迷城逃亡app详细设计说明书 |
| 当前版本： | 0.1.1 |
| 作者： | 吴子乔 |
| 完成日期： | 2018-05-13 |

# 版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 参与者 | 起止日期 | 备注 | 审核人 |
| 0.1.1 | 吴子乔 | 2018-05-13至  2018-05-16 | 对详细设计说明书进行粗略填写 | 陈栩,石梦韬 |
|  |  |  |  |  |

目录

[版本历史 2](#_Toc514270610)

[一、引言 3](#_Toc514270611)

[1.1编写目的 3](#_Toc514270612)

[1.2项目背景 3](#_Toc514270613)

[1.3参考资料 3](#_Toc514270614)

[二、总体设计 4](#_Toc514270615)

[2.1需求概述 4](#_Toc514270616)

[2.2软件结构 4](#_Toc514270617)

[三、程序描述 4](#_Toc514270618)

[3.1功能 4](#_Toc514270619)

[3.2性能 6](#_Toc514270620)

[3.3输入项目 6](#_Toc514270621)

[3.4输出项目 6](#_Toc514270622)

[3.5算法 7](#_Toc514270623)

[3.6程序逻辑 8](#_Toc514270624)

[3.7接口 27](#_Toc514270625)

[3.8存储分配 27](#_Toc514270626)

# 一、引言

## 1.1编写目的

此文档编写目的在于，使软件开发人员充分了解软件的详细设计结构及软件的整体逻辑结构。

## 1.2项目背景

项目名称

基于手机android端的捉迷藏类小游戏

任务来源

杨枨老师

项目开发者

吴子乔，石梦韬，陈栩

## 1.3参考资料

[1] 游戏《迷宫深处》开发者介绍及游戏内容 2018年4月11日

https://tieba.baidu.com/p/5366126268?red\_tag=1516710533&traceid=

[2] unity社区论坛2D游戏制作技术简单举例 2018年4月20日

http://forum.china.unity3d.com/thread-13546-1-1.html

1. unity社区学习资料及unity开发教程 2018年4月20日

http://forum.china.unity3d.com/forum.php

1. CSDN博客社区.文章.软件工程需求分析文档模板 2018年4月21日

https://blog.csdn.net/jiangcl207504/article/details/5467285

1. SiKi学院unity开发相关课程 2018年4月22日

http://www.sikiedu.com/cloud/search?q=unity%E5%85%A5%E9%97%A8

[6] 张海藩、牟永敏.《软件工程导论》-6版 北京：清华大学出版社，2013（2018.1重印）

[7] 陈洪、任科、李华杰.《游戏专业概论》 北京：清华大学出版社，2010.1

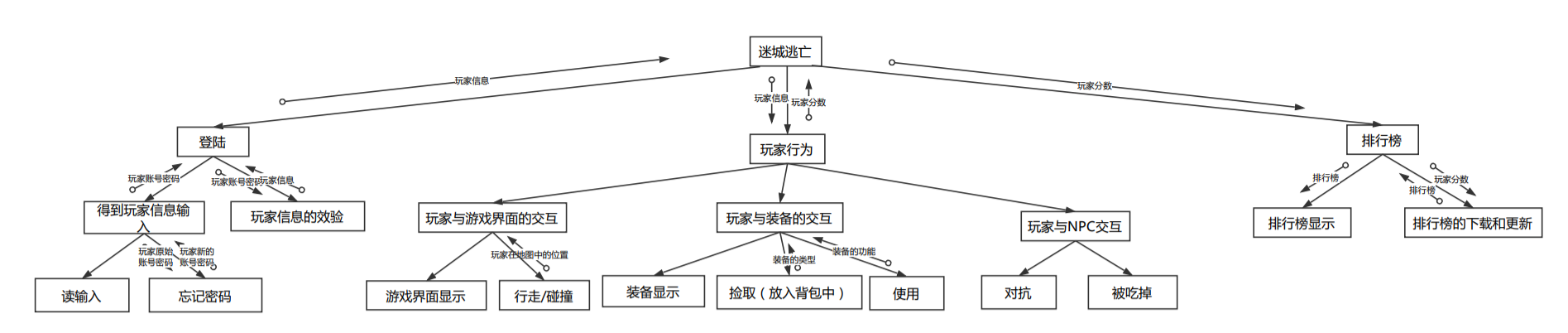
[8] ISO 9003 软件工程文档 详细设计说明书

# 二、总体设计

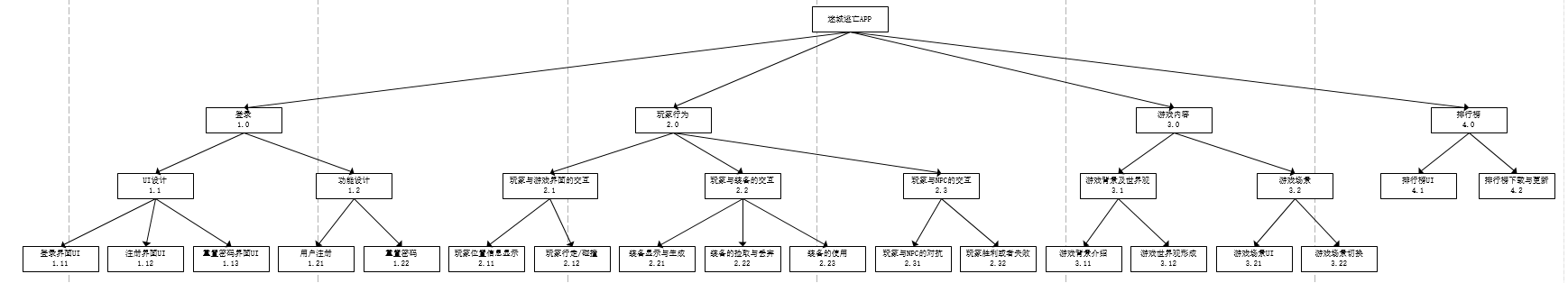
## 2.1需求概述

详见需求分析分档

## 2.2软件结构



# 三、程序描述

仅模块（1）登录及模块（3）游戏背景及世界观介绍

## 3.1功能

（1）

|  |  |
| --- | --- |
| 功能 | 功能解释 |
| 用户注册账号  用户登录  用户修改密码  播放场景动画介绍游戏背景  在每关关卡前介绍关卡背景 | 用户进行账号注册，填入账号密码等信息，程序在后台将其传入数据库储存  在用户登录时将用户输入的账号名密码等与储存在数据库中的信息进行对比，当匹配成功时允许登录  当用户忘记账号密码时，可以根据账号进行密码修改，程序在后台将修改信息上传数据库并修改数据库  当用户进入游戏时，播放游戏场景动画介绍游戏背景及游戏世界观  当用户闯入不同关卡时，有不同的关卡背景介绍 |

（3）

【玩家与游戏界面的交互（2.1）】

玩家位置信息显示（2.11）：系统能够记录下玩家在游戏当中的位置，并且玩家在游戏地图中的位置通过游戏界面的方式显示出来

玩家行走/碰撞（2.12）：玩家通过按“↑”“↓”“←”“→”箭头控制玩家的位置，当玩家与碰到碰撞快时，会产生一次碰撞，此时玩家不会按照原来箭头的方向移动，而是停留在原地。

【玩家与装备的交互（2.2）】

装备的显示与生成（2.21）：在地图非碰撞块（墙体等）的地方随机生成一些装备，供玩家在游戏的过程中和装备进行交互。

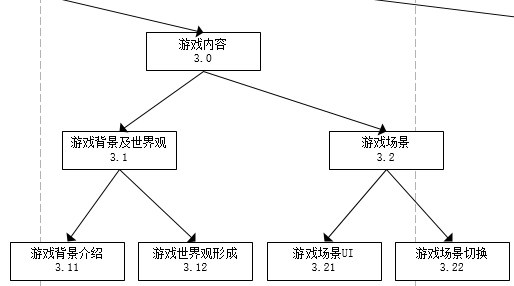
装备的捡取与丢弃（2.22）：玩家碰到地图中的道具时，假如背包容量没满，自动捡取，对于已经在背包的道具，点击道具，玩家有权决定是否使用该装备。

装备的使用（2.23）：玩家点击装备并使用时，不同类型的装备对于玩家状态有着不同的影响。

【玩家与NPC的交互（2.3）】

玩家与NPC的对抗（2.31）：人类可以和狼人进行对抗，假如人类拥有猎枪等攻击类道具，可以和狼人进行对抗，狼人可以抓到人类，增加其生命。

玩家胜利和失败（2.32）：玩家假如在游戏过程中达到了失败条件，玩家就失败，假如玩家一直坚持完成了三关游戏，玩家成功。



【游戏背景及世界观（3.1）】

（关于游戏背景介绍以及游戏世界观的形成，暂时还没有定下来，游戏背景介绍需要拥有精美的界面，这部分我们决定外包出去，该部分伪码不再展开说明）

【游戏场景（3.2）】

玩家场景UI（3.21）：游戏关卡的初始化，游戏地图的生成和游戏状态条的生成。

游戏场景切换（3.22）：一个关卡胜利之后进入到另外一个关卡，游戏失败出现排行榜界面。

## 3.2性能

## 3.3输入项目

①用户名

②密码

## 3.4输出项目

|  |  |
| --- | --- |
| 输出项目 | 项目解释 |
| 主界面  用户账号注册界面  提醒账号已存在  用户登录界面  提示用户账号错误或密码错误界面  修改密码界面  提示用户账号错误  ————————————————  密码成功修改界面  播放场景动画介绍游戏背景  在每关关卡前介绍关卡背景 | 用户进行程序时先进入主界面窗口进行功能选择  用户在主界面选择注册功能时跳往注册界面  当用户注册的账号在数据库中已有信息时，跳出提醒窗口提示用户账号已经存在  当用户选择登录窗口时，进入用户登录界面。  当用户在进行登录时账号或者密码输错时，跳出提醒窗口  当用户忘记密码点击密码修改时，进入密码修改界面  当在修改密码时用户输入的账号在数据库中不存在时，跳出提示窗口提醒用户该账号不存在  ————————————————  当用户修改密码步骤正确时，程序跳出密码成功修改界面，提示用户密码修改成功  当用户进入游戏时，播放游戏场景动画介绍游戏背景及游戏世界观  当用户闯入不同关卡时，有不同的关卡背景介绍 |

## 3.5算法

统计玩家分数 seq

游戏结束

读取当前玩家分数

显示当前玩家分数

number:=0

程序体 iter until 排行榜显示

查找排行榜中玩家信息 iter util 找到当前玩家

比较排行榜信息 select 排行榜中玩家信息==当前玩家信息

比较排行榜数据 select 排行榜中玩家分数<玩家当前分数

排行榜中玩家分数=玩家当前分数

number:=number+1

比较排行榜数据 end

比较排行榜信息 end

查找排行榜中玩家信息 end

判断number select number==0

添加玩家信息

添加玩家当前分数

判断number end

调用排序算法

上传新的排行榜信息

显示排行榜信息

程序体 end

统计玩家分数 end

## 3.6程序逻辑

(1)



如上图所示：

在模块中各个功能的调用关系。

各个功能相应的PDL伪码及N-S图：

①各个部分定义

Declare screen\_main as structure(图形界面结构) --\*定义主界面

Declare screen\_register as structure(图形界面结构) --\*定义注册界面

Declare screen\_login as structure(图形界面结构) --\*定义登录界面

Declare screen\_change as structure(图形界面结构) --\*定义修改密码界面

Declare screen\_exit as structure(图形界面结构) --\*定义离开游戏界面

Declare screen\_remb as structure(图形界面结构) --\*定义提示界面

Declare screen\_botton as structure(图形界面结构) --\*定义按钮

Declare screen\_textbook as structure(图形界面结构) --\*定义文本框输入

Declare screen\_picture as structure(图形界面结构) --\*定义图片

 ②screen\_main

Procedure screen\_main

Begin

If 玩家点击注册 then

转到screen\_register

Else if 玩家点击登录 then

转到screen\_login

Else if 玩家点击修改密码 then

转到screen\_change

Else if 玩家点击离开 then

退出游戏

End if

End

③screen\_register

Procedure screen\_register

Begin

If screen\_textbook is not null then --\*意味着用户输入了注册信息

与数据库信息进行对比

if 数据库中没有此账号 then

跳出screen\_remb提示用户注册成功

Else

If数据库中账号已经存在

跳出screen\_remb提示用户已存在账号，重新输入

 End if

End if

End if

跳转回screen\_main

End

④screen\_login

Procedure screen\_login

Begin

If screen\_textbook1 is not null then --\*先判断用户名是否存在

if 数据库中没有此账号 then

跳出screen\_remb提示用户账号不存在

Else

If screen\_textbook2 is not null then --\*再判断密码是否存在

与数据库信息进行对比

if 与数据库中的密码不匹配 then

跳出screen\_remb提示用户密码不正确

并询问用户是否忘记密码，是否要修改密码

If 用户点击修改密码 then

跳到 screen\_change

Else

重新输入密码

End if

Else

进入游戏并播放游戏动画

End if

End if

End if

End if

End

⑤screen\_change

Procedure screen\_change

Begin

If screen\_textbook1 is not null then --\*判断用户是否输入用户名

if 数据库中没有此账号 then

跳出screen\_remb提示用户账号不存在

Else

If screen\_textbook2 is not null then --\*用户输入修改密码

跳出 screen\_remb提示修改成功，转到screen\_main

End if

 End if

End if

End

 ⑥screen\_exit

Procedure screen\_exit

Begin

If 用户点击离开按钮 then

离开游戏

End if

End

(2)

|  |  |
| --- | --- |
| 模块内容： | 人类以及狼人的行走 |
| 对应HIPO图模块： | 2.1, 2.11, 2.12, 2.32 |
| PDL代码： | Procedure 人类以及狼人的行走  人类附近半径为 视野范围 的圆内，调高其附近遮罩层的亮度  //在移动界面分别点击了类似于↑↓←→的按钮  if 点击↑ then  if 玩家未触碰到障碍物 then  玩家的位置.y位置 += 角色移动速度\*间隔时间 //间隔时间在Unit3d中用time.deltatime表示  else  角色不移动  end if  else if 点击↓ then  if 玩家未触碰到障碍物 then  玩家的位置.y位置 -= 角色移动速度\*间隔时间  else  角色不移动  end if  else if 点击← then  if 玩家未触碰到障碍物 then  玩家的位置.x位置 -= 角色移动速度\*间隔时间  else  角色不移动  end if  else if 点击→ then  if 玩家未触碰到障碍物 then  玩家的位置.x位置 += 角色移动速度\*间隔时间  else  角色不移动  end if  else  角色不移动  end if    在游戏界面中显示角色对应的图像并改变人类附近遮罩层的在游戏界面位置  角色体力 = 角色体力 - 体力消耗速度 \* 间隔时间;    if 角色名=="狼人" and 角色体力 == 0  游戏状态 = false  end if    end 人类以及狼人的行走 |
| N-S结构图： | 人类及狼人的行走.jpg |

|  |  |
| --- | --- |
| 模块内容： | 玩家与装备的交互 |
| 对应HIPO图模块： | 2.2 |
| PDL代码： | Procedure 玩家与装备的交互  执行 玩家拾取装备  执行 玩家丢弃装备  执行 玩家使用装备  end 玩家与装备的交互 |
| N-S结构图： |  |

|  |  |
| --- | --- |
| 模块内容： | 玩家丢弃装备 |
| 对应HIPO图模块： | 2.22 |
| PDL代码： | Procedure 玩家丢弃装备  if 点击丢弃按钮 then  该装备丢失  else  保留该装备  end if  end 玩家丢弃装备 |
| N-S结构图： | / |

|  |  |
| --- | --- |
| 模块内容： | 玩家拾取装备 |
| 对应HIPO图模块： | 2.22 |
| PDL代码： | Procedure 玩家拾取装备  if 玩家的位置刚好在道具位置范围之内 then //玩家刚好触碰到道具，Unity3d可以用刚体碰撞实现  if 装备箱.size <= 装备总量 then  在该地方的装备消失  装备箱.add(该地方的装备) //向装备箱增加一个装备  else  装备仍然保留在原地  end if  end if  end 玩家拾取装备 |
| N-S结构图： |  |

|  |  |
| --- | --- |
| 模块内容： | 玩家使用装备 |
| 对应HIPO图模块： | 2.23, 2.31 |
| PDL代码： | Procedure 玩家使用装备  //对应游戏界面状态条点击编号，使用装备  if 玩家点击游戏状态条装备部分的对应装备 then  提取该装备的信息  declare 是否删除该装备 as bool  是否删除该装备 = true  if 该装备是猎枪 then  //装备具体功能待定  if 角色名 == "人类" then  if 人类所在区域内存在狼人 then  人类所在区域内的狼人消失  else  是否删除该装备 = false  end if  else //角色名 == 狼人 then  if 狼人附近是否存在人类 then  狼人附近的人类消失  角色体力 += (最大体力-角色体力)\*Random(0.2,0.8)  else  是否删除该装备 = false  end if  end if  else if 该装备是防狼喷雾 then  //装备具体功效待定  狼人找到玩家的概率减小  坚持10秒钟  狼人找到玩家的概率增大  else if 该装备是铲子 then  if 玩家站在障碍物附近并且玩家朝向障碍物 then  玩家朝向的障碍物消失  else  是否删除该装备 = false //该装备没有使用成功  end if  else if 该装备是照明弹 then  将整个地图的遮罩层亮度调为1  等待3秒钟  将整个地图的遮罩层亮度调为0  else if 该装备是隐身衣 then  地图界面中角色画面的透明度调为0.9  对于敌方来说角色的坐标轴位置消失  坚持10秒钟  对于敌方来说角色的坐标轴位置重现  地图界面中角色画面的透明度调为0  else if 该装备是傀儡 then  //装备功能待定  地图界面中新生成一个人物，其行动路线和原始人类行动路线相同  else if 该装备是时光药水 then  //装备的功能待定  关卡时间 += 60  else if 该装备是体力药水 then  角色体力 += (最大体力-角色体力)\*Random(0,1)  else if 该装备是手电筒 then  视野范围 += 1  end if  if 是否删除该装备 == true then  装备箱.delete(该装备)  end if  end if    end 玩家使用装备 |
| N-S结构图： | 玩家使用装备.jpg |

|  |  |
| --- | --- |
| 模块内容： | 人类以及狼人的交互 |
| 对应HIPO图模块： | 2.3, 2.32 |
| PDL代码： | Procedure 人类以及狼人的交互  if 角色名 == "人类" then //狼人刚好触碰到人类，Unity3d可以用刚体碰撞实现  if 狼人触碰到了人类  关卡界面及地图界面停止  游戏状态 = false //游戏状态用来判断游戏胜利与失败的条件  end if  else // 角色名 == "狼人" then  if 狼人触碰到了人类  角色体力 += (最大体力-角色体力)\*Random(0.2,0.8)  end if  end if    end 人类以及狼人的交互 |
| N-S结构图： |  |

|  |  |
| --- | --- |
| 模块内容： | 游戏界面的生成 |
| 对应HIPO图模块： | 3.2 |
| PDL代码： | Procedure 游戏界面的生成 is  //公共变量暂时先放在这里定义，以便所有的函数能够访问该变量  declare 关卡数 as scalar  declare 角色名 as string  declare 游戏状态 as bool    关卡数 = 1  loop while 关卡数 <= 3  执行 关卡界面的生成  关卡数++  end loop  游戏状态 = true  end 游戏界面的生成 |
| N-S结构图： |  |

|  |  |
| --- | --- |
| 模块内容： | 关卡界面的生成 |
| 对应HIPO图模块： | 3.2 |
| PDL代码： | Procedure 关卡界面的生成  展示游戏剧情界面  执行 场景界面的生成  if 游戏状态 = false then  终止关卡界面的运行  执行 游戏结束界面的生成  end if  end 关卡界面的生成 |
| N-S结构图： |  |

|  |  |
| --- | --- |
| 模块内容： | 场景界面的生成 |
| 对应HIPO图模块： | 3.2 |
| PDL代码： | Procedure 场景界面的生成  //要用到的公共变量先暂时放在这里面定义  declare 角色等级 as scalar  declare 游戏分数 as scalar  declare 角色速度 as scalar  declare 离敌方距离 as scalar  declare 玩家的位置 as struct  declare 角色体力 as scalar  declare 最大体力 as scalar  declare 视野范围 as scalar  declare 当天状态 as string  declare 关卡时间 as scalar  declare 关卡经过时间 as scalar  declare 装备总量 as scalar  declare 装备箱 as list  declare 背包已有物品 as list    执行 地图及装备的生成  执行 游戏状态条的生成  end 场景界面的生成 |
| N-S结构图： |  |

|  |  |
| --- | --- |
| 模块内容： | 游戏状态条的生成 |
| 对应HIPO图模块： | 3.21 |
| PDL代码： | Procedure 游戏状态条的生成  //游戏状态条角色的信息  print "角色"  输出角色对应的图像  print "等级"  print "分数"  print "速度"  print "离敌方距离"  //游戏状态条关卡的信息  print "关卡" + 关卡数  输出上下左右对应的图像  输出背包轮廓的图像    //对应Unity3d里面的Update()函数，系统会自动执行  loop while 游戏状态 == true  删除掉原来已经输出过的属性  获取实时属性  //所有输出都输出到游戏状态条上  //游戏状态条角色的信息  print 角色名  print 角色等级  print 游戏分数  print 角色速度  print 角色体力  //游戏状态条关卡的信息  print 关卡时间-关卡经过时间 //关卡剩余时间  print 当天状态  //游戏状态条  输出背包内已有的物品图像  end loop  end 游戏状态条的生成 |
| N-S结构图： |  |

|  |  |
| --- | --- |
| 模块内容： | 地图的生成 |
| 对应HIPO图模块： | 3.21, 2.21 |
| PDL代码： | Procedure 地图的生成  declare 地图块列表 as list //列表内存放着包括地面块，墙体块以及其它障碍块，出口块等信息  declare 地图长 as scalar  declare 地图宽 as scalar  declare 障碍块长度 as scalar //这里障碍块长度和宽度相等，方便人物上下行走速度的统一以及为了游戏的美观  //由于本游戏的特殊性，我们可以将一整个迷宫划分成若干个大的迷宫块，大的迷宫块有一条出路，并且大的迷宫块里面也有一条出路  declare 大块迷宫 as array //n\*n二维数组      //declare i as scalar  //declare j as scalar  //i = 0;  //loop while i < 地图长/障碍块长度  // j = 0;  // loop while j < 地图宽/障碍块长度  // 在游戏界面(i\*障碍块长度,j\*障碍块长度)处生成地面  // j++  // end loop  // i++  //end loop  将地图中所有的迷宫块都铺上地砖    在大块之间从左上角、左下角、右上角、右下角这四个点选择一个点作为起点  //模拟走一遍迷宫，终点为大块迷宫边缘处某一点，至少走(n+1)步到达终点。  执行 迷宫的行走  在大块迷宫内部随机生成其它障碍物  设置地图的遮罩层，将整个地图的遮罩层亮度调为0  在地图中没有障碍物的地方生成装备  end 地图的生成 |
| N-S结构图： | 地图的生成.jpg |

(3)

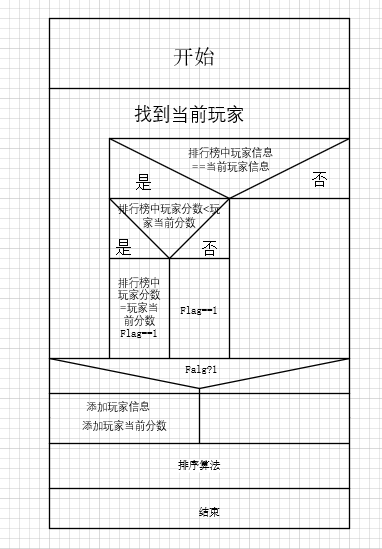
各个功能相应的PDL伪码及N-S图：

①各个部分定义

Declare screen\_ ranking as structure(图形界面结构) --\*定义排行榜界面

②screen\_ranking

procedure score

begin

struct p=排行榜;

int number=0;

loop util i<p.length

if p[i].name==name

if p[i].score<score

p[i].score=score

end if

number=1;

end if

end loop

if number==1

p =new 排行榜；

p[i].name=name;

p[i].score=score;

end if

sort();

end

## 3.7接口

与云端数据库的接口

## 3.8存储分配

在运行程序的设备上分配空间，存储云端下载的游戏数据