2019年6月21日18:47:58

首先阐述一下可能需要实现的相关需求，需求对应的具体的功能，功能产生模块的划分

初步需要实现的需求，通过整理大致如下（根据流程时序整理）

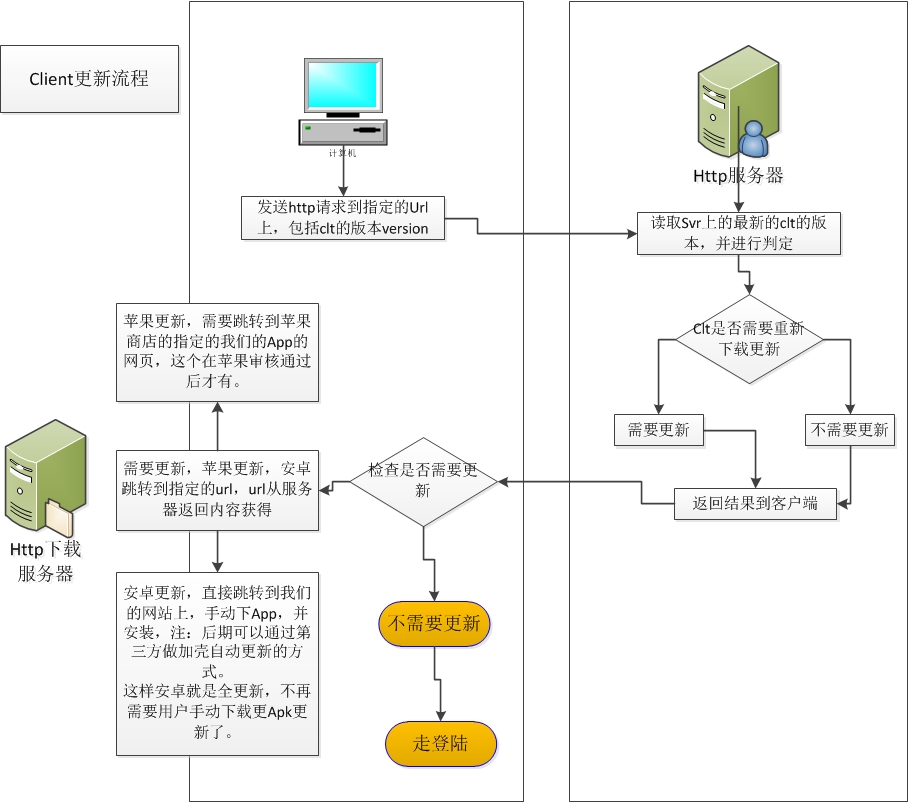
1：更新需求

说一下：由于苹果国内审核的严格带来的不便，以及，热更新可以更好及时的修正客户端中的错误，所以一般而言，都会考虑进行热更新的设计。

热更新的设计，还需要客户端的相关技术能够支持。初期是否做，待考虑。

一般而言，由于苹果的限制，热更新更加倾向于code代码更新，文本更新。资源类更新在苹果平台有比较大限制。同理，尽管安卓随时可以更新，但是，为了对齐苹果，需要考量。

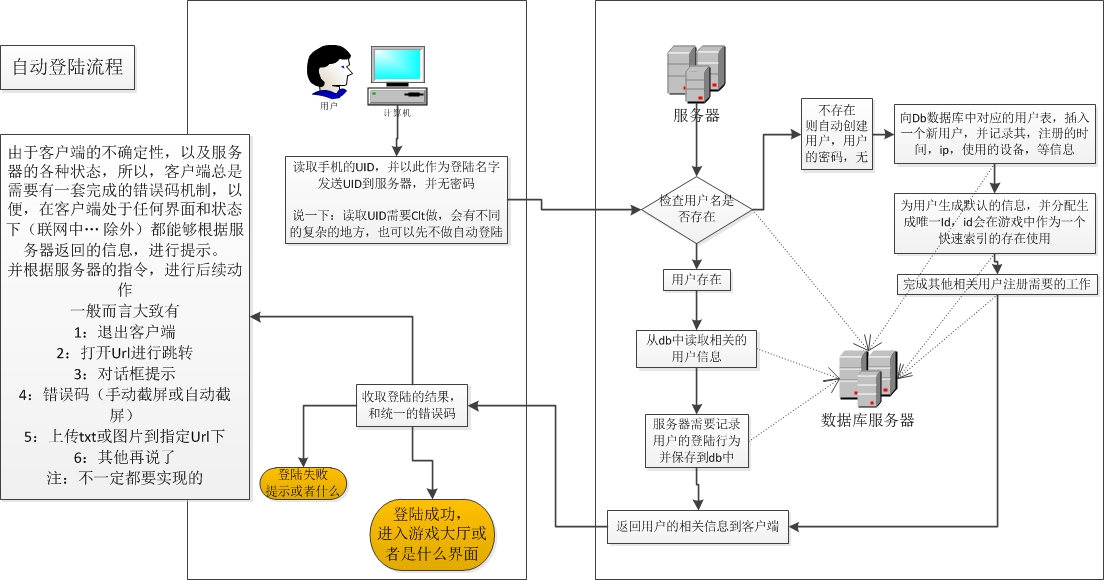
大致流程如下：



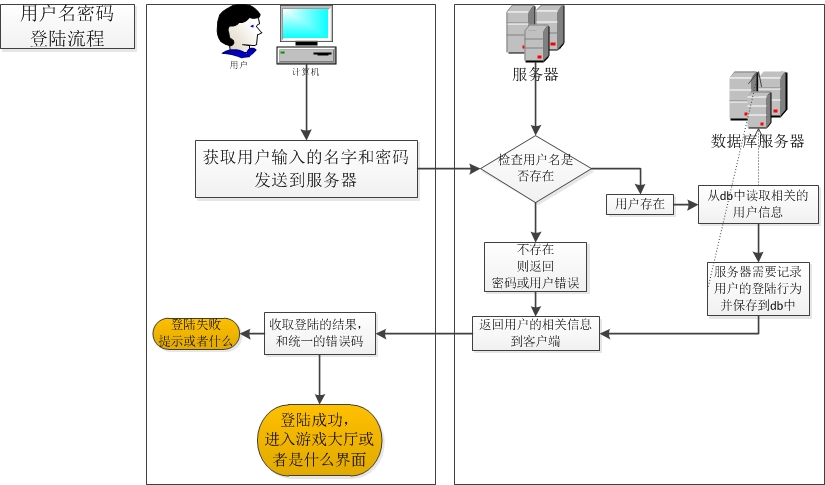
补充一下：热更新的话，再找时间商议。（热更新会导致客户端有比较大的变化）

2：登陆需求

2.1：自动登陆需求

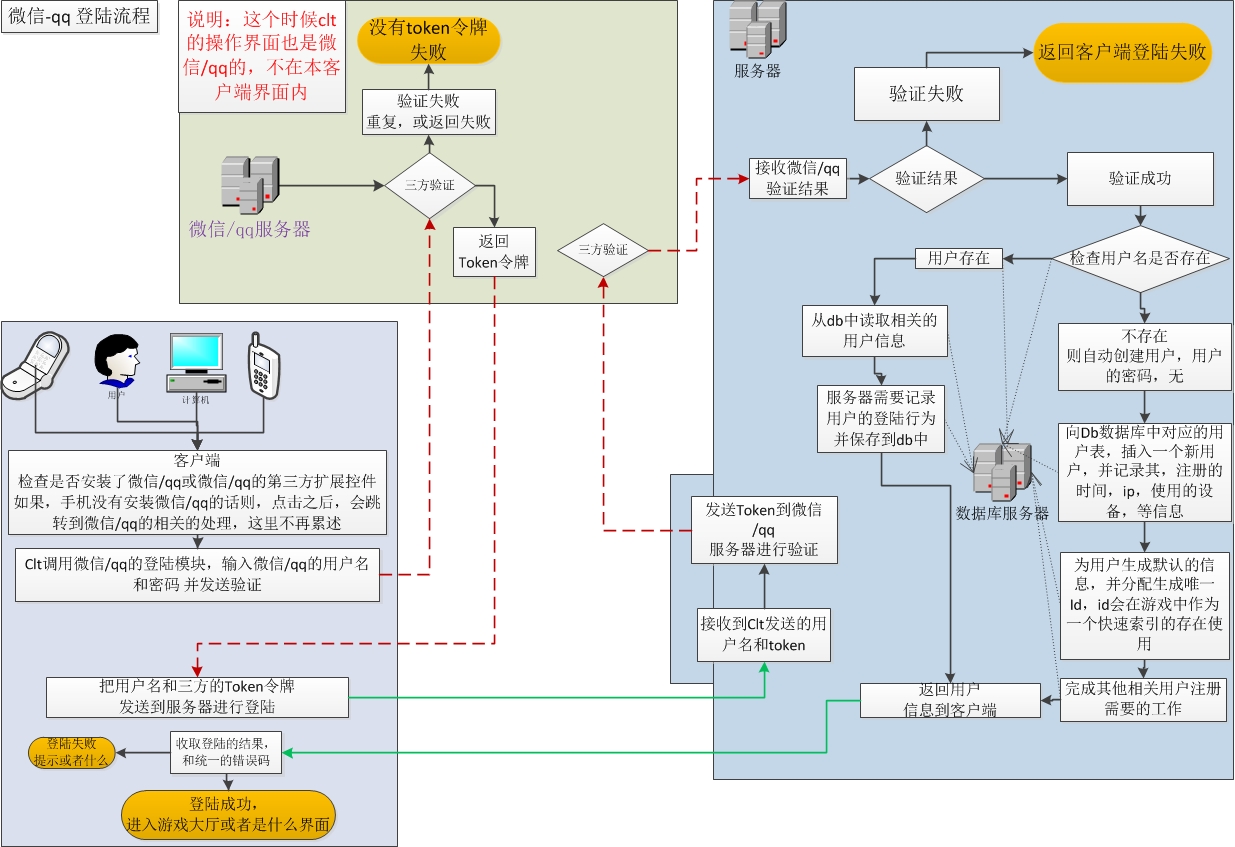


2.2：用户名，或，手机号码，或，设备id，加，登陆密码，或无登陆密码



2.3：微信第三方登陆

2.4：qq第三方登陆



3：客户端进入到游戏大厅的需求实现

说一下：客户端登陆成功后，进入到游戏大厅，可以查看，用户的物品数量，金币（暂定）数量，卡片（暂定）数量。用户可以通过，选择游戏列表，进入到具体的某一个游戏之中。游戏列表（暂定）

这里可能还会弹出每日签到，或者是游戏活动，游戏比赛，以及广播信息的滚动，或者是联系客服，浏览网站，等等相关的操作。

4：用户在客户端点击了具体的游戏

说一下：用户点击了斗地主，界面切换到斗地主的界面（也可以就在大厅），客户端的tcp连接切换到具体的游戏服务器（初期服务器都在一起，以后再说）

用户可以，进行随机匹配，开房间，单机模式（练习），教学模式（也可以没有），等等操作。

4.1：用户选择随机匹配

需要游戏服务器或者是core核心服务器，能够完成相同操作用户的匹配。当匹配无法完全实现时，需要有机器人参与匹配。

用户在匹配完成之前，可以取消匹配。在匹配完成之后，也可以退出匹配，具体的约束条件再说，可以消耗用户一次游戏的机会或者是之类的。

4.2：用户选择，开房间模式。

需要在游戏服务器上，根据用户的房间设定，进行对应的房间开设，并令用户自动进入房间，并给与用户房间id号。这个部分，可能还需要能够把房间id号，发布到微信中，或者是qq中，以方便用完成，邀约好友。

4.3：用户可以进行的其他行为，如：单机模式，则由单机来完成行为。

4.4：用户如果希望提高，游戏的能力，则提供，诸如：记牌器，推荐出牌，概率分析之类 的帮助（暂定，也可以没有）

4.5：其他再补充

5：游戏进行中

说一下：单机游戏，则游戏的相关过程逻辑，由单机来完成。

多人游戏，则需要clt客户端发送网络消息msg到svr服务器，由服务器完成逻辑，并广播到其他玩家。在多人游戏过程中，可以玩家可以进行，表情符号聊天，地方方言语音播放（暂停，也可以没有），短语音聊天，等交互行为。

再补充：

6：单次游戏结束

显示游戏结果，胜负，并进行扣分等行为。

7：整体游戏结束

随机匹配，当玩家的游戏分数不够时，则结束整体游戏。令玩家退到游戏大厅。

开房间模式，则当房间设定的游戏多少盘满足结束，或者是房间设定的游戏结束规则，比如：某一个玩家没有分数，或者打满8轮16轮，或者是，东南西北的庄家都下庄之类的。

游戏结果，如果，在微信登陆下，可以截屏分享到微信群。等等。

8：其他附带的可能的功能。

8.1：历史回放

可以回放玩家的某一个历史游戏，最多纪录16场或者8场之类的。

8.2：自己的战绩

可以统计纪录玩家的战绩。

8.3：比赛功能，排行榜（一般按照时间区分，周，月，季度之类）等等

9：补充

分割=======================线===================

目前经过相关分析，服务器部分，需要进行的相关设计，大致有（以后补齐）

1：数据库部分

1.1：数据库的打开（连接），关闭。

1.2：数据库中，表，的，建立，删除。

1.3：数据库中，表，的，相关控制。新增数据（行），删除数据（行），写入数据（行或单个或多个）

更新数据（行或单列，多列）

--以上接口总是需要的，另外和相关采用的语言（go或c++），以及相关技术有关系。

2：游戏逻辑部分

1.1：用户部分

牵扯到用户的相关数据部分，游戏的需求功能部分（如：金币，积分，房卡之类的）

用户的，名字（或UID或微信/qq的名字），头像（微信/qq有头像），等级，之类的。

1.2：更新部分

当发生服务器更新时，需要向下兼容对应的数据库中的表的相关设计（或修改表），以及相关的

服务器配置等等。

当发生客户端更新时，苹果方面需要审核通过。安卓方面，需要把最新的apk上传到http的下载url中，并，处理相关更新版本迭代。以要求客户端同步下载apk（后期考虑安卓全自动更新）

1.3：登陆部分

需要根据具体的需求，以及用户的基本信息等相关，建立对应的数据库表，以及其他的相关。

1.4：总用户功能部分

需要建立用户的总表，以做相关特定功能的设计支撑，如：全员广播，公告，活动提示，聊天，邮件（如果有的话），每日签到（如果有的话），等等。

说一下：相当于在core上，设计一个总的用户跟踪表（map或hash\_map），并利用该表，实现相关的可能的功能模块扩展等等。

1.5：游戏房间部分

游戏房间的建立，销毁。房间id的产生（可能会有房间存盘），玩家的房间进入，离开。以及房主的权利（踢人，解散，禁言等），房卡的扣除（一般是开房间锁定房卡，正式开始才扣）

房间中的聊天广播，表情，语音（如果做的话）

房间的总体管理

1.6：游戏逻辑部分

需要根据房间的设定，以及游戏的具体规则，来考虑实现。

说一下：这个部分是在客户端有单机模式下，会有编码的重复。所以，可以考虑用c++统一只做一次。（再定）

1.6.1：游戏结果以及流程控制算法。

简单说，就是需要根据不同的游戏规则，建立可能的游戏结果均衡控制的系统和模块。

避免，玩家的输赢幅度太大，引起不良效果。以及将来可能的一些提高用户粘度的地方。

1.7：匹配部分

用户随机匹配开始游戏，消耗一些什么，具体规则再定。

1.8：机器人部分

这个地方比较麻烦，需要比较复杂。再定。

3：网络部分

基本上确定使用go作为，主要的编码语言。这个部分，有go的相关底层库来完成。

便利化。