机器人开发文档

[**一. 机器人介绍** 1](#_Toc7099)

[1. 机器人整体ui如下： 1](#_Toc31328)

[1. 模块介绍 1](#_Toc27160)

[2. 任务管理 2](#_Toc22183)

[3. 添加和修改任务 3](#_Toc452)

[2. 配置文件为RobotSet.ini: 3](#_Toc582)

[1. [Static]配置项介绍，需要在启动机器人之前完成配置 3](#_Toc16032)

[2. [Dynamic]配置项介绍，在启动机器人之后进行配置，具体ui参照上方的添加和修改任务 3](#_Toc17823)

[**二. 主要流程介绍** 4](#_Toc23408)

[1. 大厅登陆流程 4](#_Toc15552)

[2. 房间登陆流程 5](#_Toc1782)

[3. 存取钱操作 6](#_Toc6799)

[4. 坐桌逻辑 6](#_Toc4870)

[5. 离桌流程 7](#_Toc12810)

[6. 启动游戏逻辑 7](#_Toc5430)

[**三. 代码目录介绍** 7](#_Toc24248)

[1. Common目录 7](#_Toc19989)

[2. UI目录 7](#_Toc28037)

[3. PlatformLogic目录 8](#_Toc4877)

[4. GameLogic目录 8](#_Toc29086)

[**四. 机器人编译** 8](#_Toc6235)

[**五. 机器人运行** 9](#_Toc3336)

# **一. 机器人介绍**

此文档主要针对从登陆大厅到进入房间坐下启动游戏逻辑的介绍。

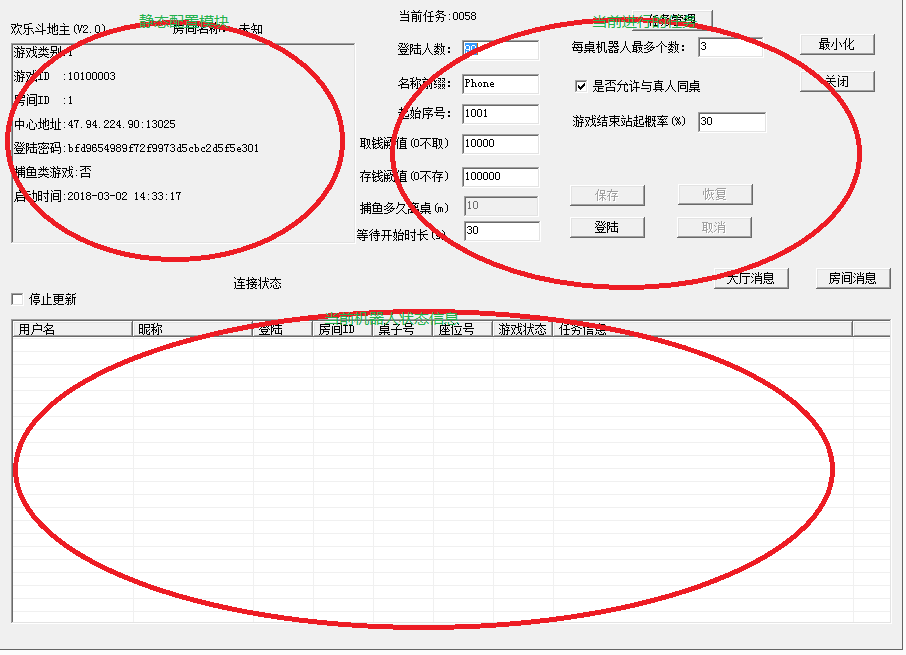
## 机器人整体ui如下：

### 1. 模块介绍

静态配置模块：启动之前配置好，机器人登陆房间的基本配置信息；

当前进行的任务：根据时段，自动切换任务，任务管理可添加不同的机器人时段任务；

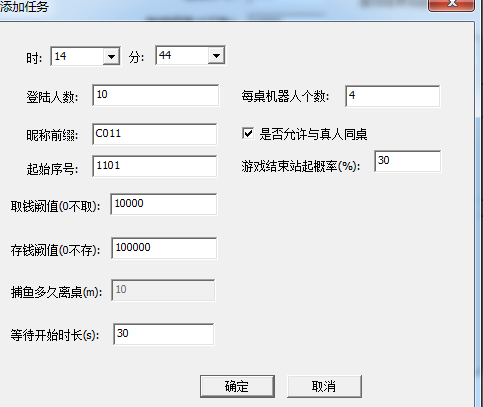
当前机器人状态信息：实时更新当前机器人的状态；



### 2. 任务管理



### 3. 添加和修改任务



## 配置文件为RobotSet.ini:

### 1. [Static]配置项介绍，需要在启动机器人之前完成配置

|  |  |  |
| --- | --- | --- |
| 配置项 | 功能 | 备注 |
| LogFile | 日志文件前缀 | 配合日期每天生成一个消息日志 |
| TitleText | 机器人标题名 | 一般配置对应的房间名 |
| KindID | 游戏类型 | 需要与游戏在平台所属的类型一致 |
| NameID | 游戏ID | 机器人需要进入的房间对应的游戏ID |
| RoomID | 房间ID | 机器人需要进入的房间对应的ID |
| CenterServerIPAddr | 前置机服务器地址 | AServer服务的地址 |
| CenterServerPort | 前置机服务器端口 | AServer服务的端口 |
| AIPWD | 机器人账号密码 | 所有机器人账号共用密码 |
| FishGame | 是否为捕鱼类游戏 | 没有准备消息，坐下即可游戏的 |

### 2. [Dynamic]配置项介绍，在启动机器人之后进行配置，具体ui参照上方的[添加和修改任务](#_添加和修改任务：)

|  |  |  |
| --- | --- | --- |
| 配置项 | 功能 | 备注 |
| 时/分 | 启动任务的开始时间 | 每天启动任务的开始时间，若无替换任务，直到凌晨才会停止 |
| 登陆人数 | 进入房间的机器人个数 |  |
| 昵称前缀 | 机器人用户名前缀 |  |
| 起始序号 | 机器人用户名后缀起始号 | 昵称与后缀合在一起即为机器人用户名，需要为数字进行加1以组合不同的机器人用户名 |
| 取钱阙值 | 当用户低于多少金币时，进行取钱 | 0表示不取，取金额:（存钱阙值-取钱阙值)/ 2 + 取钱阙值 - 当前金币 |
| 存钱阙值 | 当用户多余多少金币时，进行存钱 | 0表示不存，存金额： 当前金币 - （（存钱阙值-取钱阙值)/ 2 + 取钱阙值） |
| 捕鱼多久离桌 | 捕鱼类游戏多久进行一次结算 | 单位分钟 |
| 等待开始时长 | 坐下多久仍未开始游戏则离开桌子 | 单位秒 |
| 每桌机器人个数 | 一桌最多机器人个数 | 百人类游戏配置为180，普通游戏配置为桌上最大人数 |
| 是否与真人同桌 | 是否可与真人配桌 |  |
| 游戏结束站起概率 | 普通游戏打完一局后，多少的概率离开桌子 | 单位百分比 |

# **主要流程介绍**

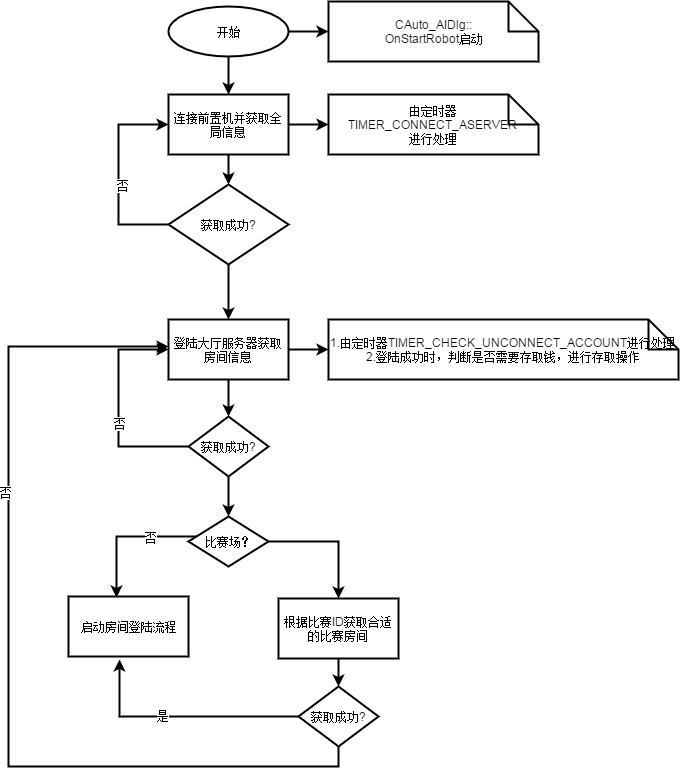
### 大厅登陆流程

注意点：

1）由定时器TIMER\_CHECK\_UNCONNECT\_ACCOUNT,每一秒钟检测是否有未登陆的账号以及是否需要切换任务；

2）登陆成功后，根据配置项判断是否需要进行[存取钱操作](#_3. 存取钱操作)；

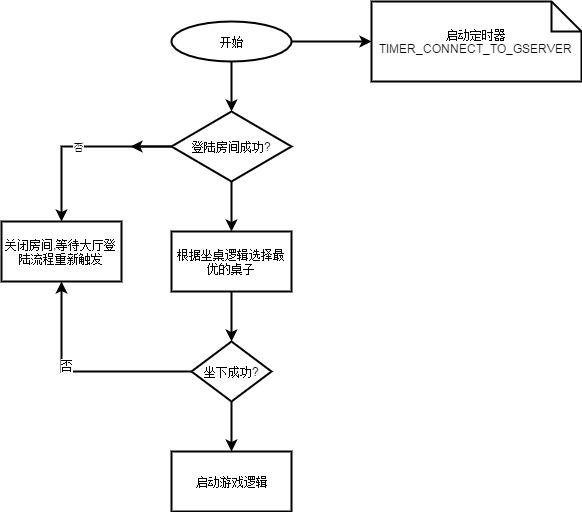
流程图如下：



### 房间登陆流程

登陆房间成功后，根据[坐桌逻辑](#_4. 坐桌逻辑)选择最优的桌子坐下。

流程图如下：



**坐下成功后，**[**启动游戏逻辑**](#_6. 启动游戏逻辑)**。**

### 3. 存取钱操作

取钱逻辑，具体[存取金额](#_[Dynamic]配置项介绍，在启动机器人之后进行配置，具体ui参照上方的添加和修改任务：)参照配置项说明:

1. 登陆大厅成功之后检测是否需要存取钱，若需要则发送存取钱操作：

接口CAuto\_AIDlg::AiControlBank()

1. 游戏结束时检测是否需要存取钱，若需要离开房间，由下次登陆大厅存取钱：

接口CGameRoomEx::CheckNeedMoneyOp()

1. 捕鱼类游戏，通过定时器TIMER\_FISHGAME\_AIBANK检测是否需要存取钱，若需要离开房间，由下次登陆大厅存取钱；

**存取钱操作成功后，将会断开与大厅和房间的链接，由定时器大厅登陆流程TIMER\_CHECK\_UNCONNECT\_ACCOUNT触发重新登陆。**

### 4. 坐桌逻辑

选桌优先级如下，接口CGameRoomEx::OnFindBestDesk(BYTE &bDeskIndex, BYTE &bDeskStation)：

1) 若不允许与真人同桌，则选择机器人最多的桌子。

2) 若允许与真人同桌，优先选择真人最多的桌子, 否则选择桌子上人数最多的桌子。

### 5. 离桌流程

离桌触发:

1) 坐下成功后，根据配置项多久未开始游戏离开桌子，启动定时 器；TIMER\_CHECK\_USERSTATE

2) 游戏结束后，根据配置项游戏结束后站起概率，启动定时器；TIMER\_CHECK\_USERSTATE

3) 捕鱼类游戏，坐下成功后，根据配置项捕鱼类游戏多久离桌，启动定时器；TIMER\_FISHGAME\_LEAVE

4) 捕鱼类游戏，由TIMER\_FISHGAME\_AIBANK定时器检测是否需要[存取钱操作](#_3. 存取钱操作)，若需要则离开桌子；

1. 游戏结束时检测是否需要存取钱，若需要离开房间；

**离桌成功后，由大厅登陆流程定时器TIMER\_CHECK\_UNCONNECT\_ACCOUNT触发重新登陆。**

### 6. 启动游戏逻辑

坐下成功后触发：

1) 创建对应的游戏逻辑处理对象，接口CGameRoomEx::StartGameClient()

2) 调用CGameRoomEx::ControlGameClient(UINT uControlCode, void \* pControlData, UINT uDataSize)启动游戏逻辑

# **代码目录介绍**

### Common目录

此目录存放一些共用的接口和实现，具体说明如下：

AIComFunction.h 一些实用的通用函数；

PlatConfig.h 单实例全局配置项类,分为[静态配置](#_1. [Static]配置项介绍，需要在启动机器人之前完成配置)和[动态配置](#_2. [Dynamic]配置项介绍，在启动机器人之后进行配置，具体ui参照上方的添加和修改任务)；

PlatglobalData.h 运行时的一些动态信息；

PlatLog 和 PlatLock，日志和锁类；

### UI目录

此目录存放当前所有机器人状态ui和任务管理ui实现：

UserStatusDialog.h 当前当前所有机器人的状态；

DialogTaskList.h 查看所有任务；

DialogTaskEdit.h 编辑单个任务；

### PlatformLogic目录

此目录存放机器人整体UI代码以及各流程逻辑实现代码：

BaseRoom.h TCP客户端连接基类, 接收和发送数据；

Auto\_Dlg.h 整体UI, 处理[大厅登陆流程](#_大厅登陆流程)和[存取钱操作](#_3. 存取钱操作)；

GameListCtrl.h 游戏房间信息管理类

Gameusermanage.h AI账号信息管理类

GameRoomEx.h 每个AI账号一个房间, 判断是否需要[存取钱操作](#_3. 存取钱操作)，处理[房间登陆流程](#_房间登陆流程)，[坐桌逻辑](#_4. 坐桌逻辑)，[离桌逻辑](#_5. 离桌流程)和[启动游戏逻辑](#_6. 启动游戏逻辑)

GameFrameDlg.h 接口头文件, 游戏逻辑需要继承IFrameInterface实现。

### 4. GameLogic目录

具体游戏代码文件, 编译机器人之前，需要将对应机器人的逻辑代码拷贝到此目录。

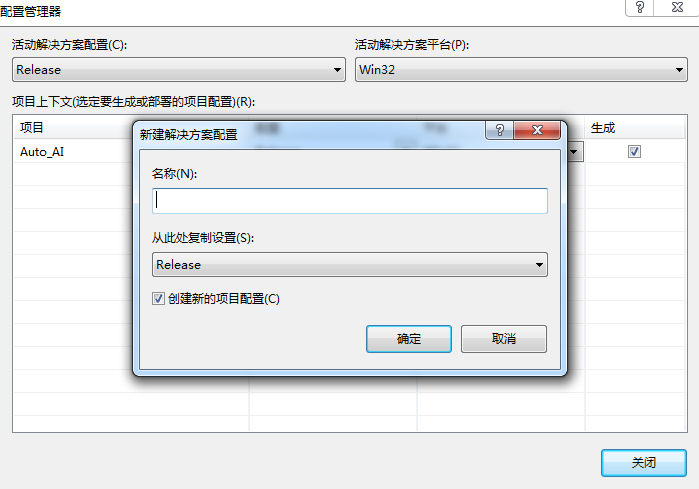
# **机器人编译**

具体步骤如下：

1. 拷贝具体游戏代码文件到GameLogic目录（如果目录下有文件需要先删除）
2. 打开工程Auto\_AI.sln, 将GameLogic目录下的文件加载到GameLogic筛选器下（如果原先加载过其他的游戏代码文件，需要先删除筛选器下的文件）



1. 新建解决方案（从Release生成）（此步骤可选）



1. 编译生成

# **机器人运行**

具体步骤如下：

1. 将ini目录下的RobotSet.ini拷贝到生成的机器人目录下

2. 打开Platform/client目录下的工程ClientPlatform.sln,生成AfcSocket.dll



3. 拷贝AfcSocket.dll到生成的机器人目录下

4. 对RobotSet.ini进行配置

6. 运行机器人