

AP_PROJ1-CAP

a Commodity Auction Platform 一个商品竞拍平台

repo: https://github.com/yunzinan/AP_CAP (https://github.com/yunzinan/AP_CAP)

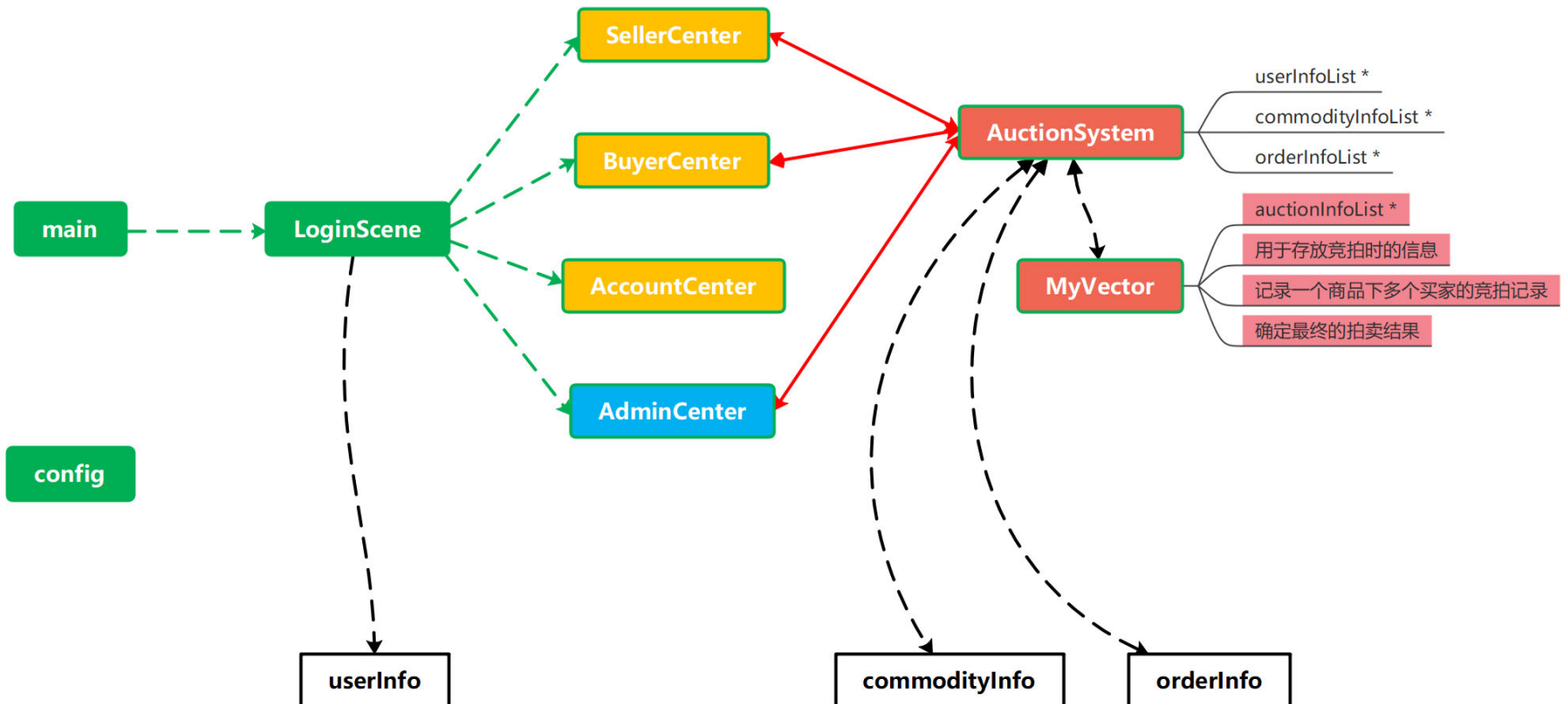
211850009 欧阳坤

PS: already finished v1.0 and will add GUI to it.

Module Design

Overview

Module Design



Module Intro

- **main**: entry of the program
- **config**: #define s
- **LoginScene**: operating interface, the main menu
- **BuyerCenter**, **SellerCenter**, **AccountCenter**, **AdminCenter**: operating interfaces, the secondary menus, interact with the **core**.

- AuctionSystem: **core**, processing orders, commodities, etc.
- MyVector: customized data structure, dealing with the auctionInfo
- UserInfo, CommodityInfo, OrderInfo: datas, .txt files

- LoginScene

```
-----  
*****LOGIN*****  
-----
```

```
1. UserLogin 2. UserRegister 3. AdminLogin 4. exit  
type a number to continue: 1  
-----
```

```
*****USER LOGIN*****  
-----
```

```
username: alice  
password: 123  
success!  
-----
```

```
*****User Menu*****  
-----
```

```
Welcome, alice  
1. Seller Center 2. Buyer Center 3. Account Center 4. exit  
select a num: _
```

- SellerCenter

*****User Menu*****

Welcome, alice

1. Seller Center 2. Buyer Center 3. Account Center 4. exit

select a num: 1

going to the Seller Center...

*****Seller Center*****

1. Release Commodity 2. View Commodity List

3. Modify Commodity 4. Cancel Commodity 5. View Order List 6. Exit

type a number to continue: 2

*****Commodity List*****

ID	Name	FloorPrice	Number	AddedDate	State
----	------	------------	--------	-----------	-------

M001	apple	100.0	0	2022-09-13	removed
------	-------	-------	---	------------	---------

M002	bed	9.9	0	2022-09-13	removed
------	-----	-----	---	------------	---------

M003	shit	0.0	97	2022-09-13	onAuction
------	------	-----	----	------------	-----------

M005	computer	999.9	0	2022-09-13	removed
------	----------	-------	---	------------	---------

*****Seller Center*****

1. Release Commodity 2. View Commodity List

3. Modify Commodity 4. Cancel Commodity 5. View Order List 6. Exit

type a number to continue: _

- BuyerCenter

```
*****Seller Center*****
1. View CommodityList 2. Search Commodity
3. View DetailedCommodity 4. Bid Commodity
5. View Order List 6. Exit
type a number to continue: 3
please type in the commodityID you want to view: M002
*****
commodityID: M002
commodityName: bed
floorPrice: 9.9
number: 0
description: bed
*****

*****Seller Center*****
1. View CommodityList 2. Search Commodity
3. View DetailedCommodity 4. Bid Commodity
5. View Order List 6. Exit
type a number to continue: 4
please type in the commodityID you want to bid: M002
*****
commodityID: M002
commodityName: bed
floorPrice: 9.9
number: 0
description: bed
*****
do you want to bid ?(y/n): y
please type in your bid price: 10.0
bid success!
*****

*****Seller Center*****
1. View CommodityList 2. Search Commodity
3. View DetailedCommodity 4. Bid Commodity
5. View Order List 6. Exit
type a number to continue:
```

- Account Center

```

*****User Menu*****
Welcome, alice
1. Seller Center 2. Buyer Center 3. Account Center 4. exit
select a num: 3
going to the Account Center...

*****Account Center*****

1. View UserInfo 2. Modify UserInfo 3. Recharge 4. exit
type a number to continue: 1
*****
userID: U001
username: alice
password: 123
phoneNumber: 123123
address: NJUCS
balance: 2111.0
userState: active
*****

*****Account Center*****

1. View UserInfo 2. Modify UserInfo 3. Recharge 4. exit
type a number to continue: 2
1. username 2. password 3. phoneNumber 4. address
please type in the index of the property you want to modify: 1
your current username:alice
new username: bob
failed! username already used! try another name!

*****Account Center*****

1. View UserInfo 2. Modify UserInfo 3. Recharge 4. exit
type a number to continue:

```

- AdminCenter

```

3. Cancel Commodity 4. View OrderList
5. View UserList 6. Ban User
7. End Auction 8. Exit
type a number to continue: 7
are you sure to end auction? (y/n)
y
-----
*****Admin Center*****
-----
1. View CommodityList 2. Search Commodity
3. Cancel Commodity 4. View OrderList
5. View UserList 6. Ban User
7. End Auction 8. Exit
type a number to continue: 4
-----
*****Order List*****
-----
OrderID CommodityID SellerID BuyerID BidTime BidPrice State
T001 M001 U001 U002 2022-09-13-23:08:43 100.0 failed
T002 M002 U001 U002 2022-09-13-23:09:02 10.0 failed
T003 M003 U001 U002 2022-09-13-23:09:14 0.0 succeeded
T004 M001 U001 U003 2022-09-13-23:09:50 101.0 succeeded
T005 M002 U001 U003 2022-09-13-23:10:00 20.0 succeeded
T006 M003 U001 U003 2022-09-13-23:10:12 0.0 succeeded
T007 M004 U003 U001 2022-09-13-23:14:04 10.2 failed
T008 M004 U003 U002 2022-09-13-23:15:16 10.2 failed
T009 M003 U001 U002 2022-09-13-23:33:13 1.0 succeeded
T010 M005 U001 U002 2022-09-13-23:33:36 999.9 succeeded
T011 M005 U001 U003 2022-09-13-23:34:47 999.9 failed
T012 M005 U001 U002 2022-09-13-23:39:33 999.9 failed
T013 M005 U001 U003 2022-09-13-23:42:36 1000.0 succeeded
T014 M002 U001 U001 2022-09-20-19:32:13 10.0 failed
-----
*****Admin Center*****
-----
1. View CommodityList 2. Search Commodity
3. Cancel Commodity 4. View OrderList
5. View UserList 6. Ban User
7. End Auction 8. Exit
type a number to continue:

```

Data Structure

Overview

- userInfo(with I/O)
- commodityInfo(with I/O)
- orderInfo(with I/O)

dynamically created when `loadInfo` function is called, deleted when `exit` with the object.

so that the same `infoList` will be created only once, and accessed indirectly via the *pointer*, which will transfer between classes.

- [ex]auctionInfo(created when `admin` **end the auction**, deleted when finished, *without* I/O)

- **userInfo**

```
//config.h
typedef struct userInfo
{
    std::string userID; // U001
    std::string username; // no more than 10
    std::string password; // no more than 20
    std::string phoneNumber; // no more than 40, numbers only
    std::string address; // no more than 40,
    float balance; // set precision 1
    std::string userState; // activate, inactivate
}userInfo;
```

userID, username, password, phoneNumber, address, balance, userState
U001, alice, 123, 123123, NJUCS, 2121.0, active
U002, bob, 123, 32131, NJUChem, 1008.2, active
U003, cat, 123, 222333, NJU502, 89.3, active

- **commodityInfo**

```
//config.h
typedef struct commodityInfo{
    std::string commodityID;
    std::string sellerID;
    std::string commodityName;
    std::string description;
    std::string addedDate; // yyyy-mm-dd
    std::string state; // onAuction, removed
    float floorPrice; // 底价
    int number;
}commodityInfo;
```

commodityID, commodityName, floorPrice, number, description, sellerID, addedDate, state
M001, apple, 100.0, 0, the-only-one, U001, 2022-09-13, removed
M002, bed, 9.9, 0, bed, U001, 2022-09-13, removed
M003, shit, 0.0, 97, nothing, U001, 2022-09-13, onAuction
M004, Cat, 10.1, 0, the-only-cat, U003, 2022-09-13, removed
M005, computer, 999.9, 0, pc, U001, 2022-09-13, removed

- orderInfo

```
//config.h
typedef struct orderInfo{
    std::string orderID;
    std::string commodityID;
    std::string sellerID;
    std::string buyerID;
    std::string bidTime; //yyyy-mm-dd-hh-mm-ss
    float bidPrice;
    std::string state; // succeeded, failed, inProcess, cancelled
}orderInfo;
```

```
userID, username, password, phoneNumber, address, balance, userState
U001, alice, 123, 123123, NJUCS, 2121.0, active
U002, bob, 123, 32131, NJUChem, 1008.2, active
U003, cat, 123, 222333, NJU502, 89.3, active
```

- [extra]auctionInfo

a class customized to reorganize the auction data & determine to final results.

EXAMPLE

commodityID: M001 number: 2 floorPrice: 10.0

- 1 buyerID: U002 bidTime: ... bidPrice: 10.1
- 2 buyerID: U004 bidTime: ... bidPrice: 11.1
- 3 buyerID: U003 bidTime: ... bidPrice: 10.0
- ...

```
//MyVector.h
typedef struct orderNode{
    userInfo * buyer;
    orderInfo * order;
}orderNode;

class MyVector {
```

```
commodityInfo * commodity;
userInfo * seller;
int num = 0; //commodity->number
orderNode orderList[USERCAPACITY];
int idx = 0;
public:
void ini(commodityInfo *_commodity, userInfo *_seller);
void addNode(orderInfo *_order, userInfo *_buyer);
void swap(int i, int j); //比较两个竞拍者的优先度
void rank(); //给出最后的排名
bool timeLess(std::string &time1, std::string &time2);
void update(); //确定succeeded和failed 处理后事
bool empty();
void clear();
};
```

Core functions

see [the source code \(https://github.com/yunzinan/AP_CAP\)](https://github.com/yunzinan/AP_CAP).

Thanks