1. Protocol

Depending on the requirements, it needs to switch between English and Dutch auctions.An English auction is a sale-up auction and a Dutch auctioneer a sale-down auction.A node needs to be preset for a British auction to be converted to a Dutch auction.Sets a maximum price based on the reserve price, and when the British auction reaches this preset value, it is converted to a Dutch auction.A timeline should also be set and converted into a Dutch auction if the preset maximum value is not reached before that time.The Dutch auction shall preset a reserve price. When the bid price is lower than the reserve price, the auction shall be terminated and the goods shall not be sold.(since the goods auctioned are tickets, tickets will become useless after missing a flight, and there is no sense in failing to auction, so the reserve price is set at zero)

2. Massage

There are three conditions that need to be considered in the communication of messages: the message sent by the auctioneer, the message sent by the buyer, and the general message.

The first is the information that the auctioneer sends and the buyer receives.

0: start the auction

1: given the item ID and price, it will be sent at the beginning of each round and at the time of the buyer's request

2: accept the bid

3: reject the bid

4: the auction is established, and the winner information and price are sent

5: failed auction, send failed auction product ID

6: auction ends

Second, the buyer sends and the auctioneer receives

10: to view the current item, send a message asking the auctioneer to return the item ID and the current price

11: bids, bids for goods, should be accepted or rejected by the auctioneer, only for British auctions

12: accept price, Dutch auction price only

Third, universal information

20: error message, received error message, such as ID mismatch.

3. Auctioneer

Auctioneers and each bidder have their own message attribute for receiving messages from other agents. This information is processed and cleared at each turn. At the beginning of the auction, the auction will announce the beginning of the auction, and sent to all the bidder product information. At the beginning of each round, the auctioneer also processes a markup or converts it into a Dutch auction based on the previous call. In Dutch auction, if there is a bidder to accept the bid, can clinch a deal to send a winning message to the winner.

```
class Auctioneer():
    def __init__(self,ID):
        self.ID
        self.message
        self.commodity
        self.auction
        auctioneer_ID.add(self)
```

Initialize the auctioneer

```
def send(self,receivers,type,commodity,price):
    receivers.massage.add(Massage(self,type,commodity,price))
```

The send message function records the type of message, the commodity message, and the sender message

```
def EnglishAuction(self,commodity):
        if message.type == 11:
            if message.price > acceptPrice:
                if turnWinnerID != 0:
                    self.send(turnWinnerID,3,commodity,message.price)
                acceptPrice = message.price
                turnWinnerID = massage.sender
            else:
                self.send(message.sender,3,commodity,acceptPrice)
    if turnWinnerID != 0 and acceptPrice >= int(commodity.currentPrice*1.05):
        for agent in bidderAgent_ID:
            self.send(agent,2,commodity,acceptPrice)
        commodity.currentPrice = acceptPrice
    elif turnWinnerID != 0:
        for agent in bidderAgent_ID:
            self.send(agent, 2, commodity, int(commodity.currentPrice*1.05))
        return False
    return True
```

English auction, will according to the bidder's reply to the message to increase the price or into Dutch auction.

```
def DutchAuction(self,commodity):
    for massage in self.massage:
        if massage.type == 12:
            self.send(massage.sender,4,commodity,commodity.currentPrice)
            return True
    return False
```

Dutch auction, gradually reduced price, when there is a bidder to accept the price, reply to auction success and terminate the auction

```
def Auction_Commodity(self):
    for commodity in commodityScale:
        for agent in bidderAgent_ID:
            self.send(agent, 0, commodity, commodity.initialPrice)
        self.auction = 1
        for t in range(totalTime,transTime,-1):
            for agent in bidderAgent_ID:
                self.send(agent, 1, commodity, commodity.currentPrice)
            for agent in bidderAgent_ID:
                agent.receive_massage()
            if self.EnglishAuction(commodity) == False:
```

```
                    transTime = t - 1
                    self.massage.clear()
                    break
            self.massage.clear()
        for t in range(transTime,0,-1):
            for agent in bidderAgent_ID:
                self.send(agent, 1, commodity, int(commodity.currentPrice*0.9))
            for agent in bidderAgent_ID:
                agent.receive_massage()
            if self.DutchAuction(commodity) == True:
                break
            else:
                self.massage.clear()
```

Combined with the auction, the first British auction gradually raised the price, when no one bid or a certain time after the conversion into Dutch auction.

4. Bidder

Bidder after receiving the auctioneer's bid, will check the auction is the British auction or Dutch auction. Different bidding strategies are used in different auctions. In the British auction, bidder will according to the bidding in a certain proportion of the random bid, but the final bid will not exceed their residual amount. In the Dutch auction, bidder once found that the price given in their own affordable range will immediately accept the bid.

```
class Agent():
    def __init__(self,ID):
        self.ID
        self.money = random.randint(30,100)
        self.message
        self.commodity
        bidderAgent_ID.add(self)
```

Initialize the Bidder

```
    def send(self,type,commodity,price):
        for auctioneer in auctioneer_ID:
            auctioneer.massage.add(Massage(self,type,commodity,price))
            #print(self.ID, price)
```

The send message function records the type of message, the commodity message, and the sender message

```
    def receive_massage(self):
        for massage in self.massage:
            if massage.type == 0
            elif massage.type == 1:
                self.bid(massage.commodity,massage.price)
                #print(self.ID, " receive price.")
            elif massage.type == 2:
                #print(self.ID, " price be accepted.")
            elif massage.type == 3:
```

```
                #print(self.ID, " price be refused.")
            elif massage.type == 4:
                self.money = self.money - massage.price
                self.commodity.add(massage.commodity)
            else:
                print("Bidder ",self.ID,": Error!")
        self.massage.clear()
```

Process incoming messages and decide whether or not to follow the auctioneer's bid

```
    def bid(self,commodityID,price):
        for auctioneer in auctioneer_ID:
            if auctioneer.auction == 1:
                rate = random.random()* 0.3
                if price <= self.money:
                    if int(price*rate) <= self.money:
                        self.send(11,commodityID,int(price*(1+rate)))
                    else:
                        self.send(11,commodityID,self.money)
            elif auctioneer.auction == 0:
                if price < self.money:
                    self.send(12,commodityID,price)
```

The asking price function is based on whether the British auction or Dutch auction is going on now. The British auction will increase the price according to the random proportion, but the price will not exceed its own reserve. The Dutch auction will choose to follow or not follow the price according to its reserve.

5. Implement

Define messages and items as classes, and save the collection of auctioneer, Bidder, and items in globalvariable.py, and use other files to call the environment.

Implement auctioneer and Bidder for file classes.

Start with get_start.py. The “get_start.py” files include initialization auctioneer, Bidder, and Bidder for items. The number of items and Bidder can be set artificially. I'm going to set them all to 3.