Doreen Hakimi
CS201 Restaurant V2.1
Fall 2013

DATA

```
enum state = { DoingNothing, WaitingInRestaurant, BeingSeated, Seated, ReadyToOrder,
Ordered, Eating, DoneEating, Leaving, Gone }
enum event = { none, gotHungry, followHost, seated, iKnowWhatIWant, waiterIsHere,
foodIsHere, doneEating, doneLeaving, restaurantIsFull, arrivedAtCashier, gotReceipt}

//some extra events/ states added on to deal with gui. not used in design but added on for my
reference

Host host;
Waiter waiter;
Cashier cashier;

String choice;
Menu myMenu;

Check myBill=null;

boolean stayOrLeave;
double wallet; //determined by gui
double debt;

//NOTE: check is an outer class accessible by all agents
class Check {
        int tablenum;
        double BillAmnt;
        double CustomerPaid;
        String choice;
        Waiter waiter;
        String name;
        Customer customer;
        enum State = {processing, processed, sent, paid };
}
```

MESSAGES

```
FollowMe(Menu m, Waiter w) {
        myMenu=menu;
        waiter=w;
        event=followHost;
}
WhatWouldYouLike() {
        event=waiterIsHere;
}
DoneChoosing() {
        event=iKnowWhatIWant;
}
NoRoomForYou() {
        event=restaurantIsFull;
}
HereIsYourBill(int tnum, double d) {
        myBill = new Check(choice, tnum, d);
}
HereIsYourReceiptAndChange(double num)  {
        wallet+=num;
        event=gotReceipt;
}
```

SCHEDULER

```
if state= DoingNothing and event=gotHungry
        then goToRestaurant();
if state = WaitingInRestaurant and event= followHost
        then SitDown();
if state==WaitingInRestaurant and event=restaurantIsFull
        then ShouldIStayOrShouldIGo();
if state=BeingSeated and event=seated
        then DecideWhatIWant(menu);
if state=Seated and event=iKnowWhatIWant
        then CallWaiter();
if state=ReadyToOrder and event=waiterIsHere
        then GiveOrder();
if state=Ordered and event=foodIsHere
        then EatFood();
if state=Eating and event=doneEating
        then leaveTable();
if state=GoingToCashier and event=arrivedAtCashier
        then PayMyBill();
if(state=paid and event=gotReceipt
        then LetsBounce(); // lol
if state=Leaving and event=doneLeaving
        then ClearState();
```

ACTIONS

```
goToRestaurant() {
    state=WaitingInRestaurant;
    host.IWantFood();
}
ShouldIStayOrShouldIGo() {
    if(!stayOrLeave) {
        host.IDontWantToWait();
        DoLeaveRestaurant();
        state=Leaving;
    }
    else
        event=none;
}
SitDown() {
    state=BeingSeated;
}
DecideWhatIWantToEat(Menu myMenu) {
    state=Seated;
    timer.start() {
        msgDoneChoosing();
    }
}
CallWaiter() {
    state=ReadyToOrder;
    waiter.ImReadyToOrder();
}
GiveOrder() {
    state=Ordered;
    waiter.HereIsMyChoice(choice);
    if(myMenu.getPrice(choice)>wallet) {
        String ch=myMenu.MostExpensiveICanAfford(wallet);
        if(ch==”None”)
            if(!stayOrLeave)
                waiter.CantAffordNotStaying();
                DoLeaveRestaurant();
                state=Leaving;
                return;
        else
            choice=ch;
    }
    waiter.HereIsMyChoice(choice);
    DoDisplayOrderCard();
}
```

Doreen Hakimi
CS201 Restaurant V2.1
Fall 2013

```
GiveNewOrder() {
    if(!stayOrLeave) {
        waiter.CantAffordNotStaying();
        DoLeaveRestaurant();
        state=Leaving;
        DoHideOrderCard();
        return;
    }
    ch=myMenu.OutOf(choice, wallet);
    if(ch==”None”)
        waiter.CantAffordNotStaying();
        DoLeaveRestaurant();
        state=Leaving;
        DoHideOrderCard();
        return;
    waiter.HereIsMyChoice(choice);
    event=none;
}
EatFood() {
    state=Eating;
    timer.start() {
        event=doneEating;
    }
}
leaveTable() {
    state=leaving;
    waiter.DoneEatingAndLeaving();
    DoGoToCashier();
}
PayMyBill() {
    myBill.CustomerPaid(wallet+debt);
    myBill.CustomerPaid=wallet+debt;
    debt=0;
    cashier.HereIsAPayment(myBill.tablenum, myBill.CustomerPaid);
    if(myBill.BillAmnt+debt>wallet)
        debt+=myBill.BillAmnt+debt-wallet;
    wallet=0;
    state=Paid;
}
LetsBounce() { //lol
    DoLeaveRestaurant();
    state=Leaving;
    wallet+=(random number);
}
clearState() {
        state=DoingNothing;
}
```