// 0: IamHungry()

**public** **void** gotHungry() { //from animation

print("I'm hungry");

event = AgentEvent.*gotHungry*;

stateChanged();

}

// 1: IWantFood(customer)

**public** **void** msgIWantFood(CustomerAgent cust) {

waitingCustomers.add(cust);

Do(cust + " is added to the waiting list");

stateChanged();

}

// 2: SitAtTable(customer, table)

**public** **void** msgSitAtTable(CustomerAgent customer, Table table) {

//state = AgentState.Serving;

//state = AgentState.Waiting;

print("Received msgSitAtTable from the host");

MyCustomers.add(**new** MyCustomer(customer, table));

stateChanged();

}

// 3: FollowMe(menu)

**public** **void** msgFollowMe(Set<String> menu) {

**this**.menu = menu;

event = AgentEvent.*followHost*;

stateChanged();

}

// 4: ReadyToOrder(customer);

**public** **void** msgReadyToOrder(CustomerAgent customer) {

**for**(**int** i=0; i < MyCustomers.size() ; i++) {

**if**(MyCustomers.get(i).c == customer) {

MyCustomers.get(i).state = MyCustomer.CustState.*readyToOrder*;

stateChanged();

**break**;

}

}

}

// 5: WhatWouldYouLike()

**public** **void** msgWhatWouldYouLike() {

event = AgentEvent.*makeOrder*;

stateChanged();

}

// 6: HereIsMyChoice(customer, choice)

**public** **void** msgHereIsMyChoice(CustomerAgent customer, String choice) {

**for**(**int** i=0; i < MyCustomers.size() ; i++) {

**if**(MyCustomers.get(i).c == customer) {

MyCustomers.get(i).state = MyCustomer.CustState.*waitingFood1*;

MyCustomers.get(i).choice = choice;

state = AgentState.*Waiting*;

stateChanged();

**break**;

}

}

}

// 7: HereIsAnOrder(order);

**public** **void** msgHereIsAnOrder(Order order) {

print("received an order");

orders.add(order);

stateChanged();

}

// 8: OrderIsReady(order)

**public** **void** msgOrderIsReady(Order order) {

**for**(MyCustomer cust : MyCustomers) {

**if**(cust.c == order.customer) {

cust.state = MyCustomer.CustState.*foodIsReady*;

stateChanged();

**break**;

}

}

}

// 9: HereIsYourOrder()

**public** **void** msgHereIsYourOrder() {

event = AgentEvent.*getFood*;

stateChanged();

}

// 10: IAmDone(customer)

**public** **void** msgLeavingTable(CustomerAgent customer) {

**for**(MyCustomer cust : MyCustomers) {

**if**(cust.c == customer) {

cust.state = MyCustomer.CustState.*doneEating*;

stateChanged();

**break**;

}

}

}

// 11: TableIsCleared(table)

**public** **void** msgTableIsCleared(Table table) {

print("table #" + table.tableNumber + " is cleared");

table.setUnoccupied();

stateChanged(); // so that when a customer leaves, host will check availability of tables again

}