

CUSTOMER AGENT

Data

```
private String name;
private String order;
private int hungerLevel;
Timer timer = new Timer();
private CustomerGui customerGui;
private Menu menu;

private HostAgent host;
private WaiterAgent waiter;

public enum AgentState
{DoingNothing, WaitingInRestaurant, BeingSeated, Seated, Ordered, Eating, DoneEating,
Leaving};
private AgentState state;

public enum AgentEvent
{none, gotHungry, followWaiter, seated, beingHelped, gotFood, doneEating,
doneLeaving};
AgentEvent event;
```

Messages

```
public void gotHungry() { //from animation
    print("I'm hungry");
    event = AgentEvent.gotHungry;
    stateChanged();
}

public void msgFollowMeToTable(WaiterAgent w, Menu m) {
    this.waiter=w;
    this.menu = m;
    print("Received msgSitAtTable");
    event = AgentEvent.followWaiter;
    stateChanged();
}
```

```

public void msgAnimationFinishedGoToSeat() {
    //from animation
    event = AgentEvent.seated;
    stateChanged();
}
public void msgAnimationFinishedLeaveRestaurant() {
    //from animation
    event = AgentEvent.doneLeaving;
    stateChanged();
}
public void msgHereForOrder() {
    event = AgentEvent.beingHelped;
    stateChanged();
}

public void msgDeliveredFood() {
    event=AgentEvent.gotFood;
    stateChanged();
}

```

Scheduler

- 1.) if (state == AgentState.DoingNothing && event == AgentEvent.gotHungry){
 state = AgentState.WaitingInRestaurant;
 goToRestaurant();
 return true;
 }
- 2.) if (state==AgentState.WaitingInRestaurant &&event == AgentEvent.followWaiter)
 {
 state = AgentState.BeingSeated;
 SitDown();
 return true;
 }
- 3.) if (state == AgentState.BeingSeated && event == AgentEvent.seated){
 state = AgentState.Seated;
 PickUpAndPeruseMenu();
 return true;
 }

- 4.)

```
    if (state == AgentState.Seated && event == AgentEvent.beingHelped) {  
        //print("In customer?");  
        state=AgentState.Ordered;  
        OrderFood();  
        return true;  
    }
```
- 5.)

```
    if (state == AgentState.Ordered && event == AgentEvent.gotFood) {  
        state = AgentState.Eating;  
        EatFood();  
        return true;  
    }
```
- 6.)

```
    if (state == AgentState.Eating && event == AgentEvent.doneEating){  
        state = AgentState.Leaving;  
        leaveTable();  
        return true;  
    }
```
- 7.)

```
    if (state == AgentState.Leaving && event == AgentEvent.doneLeaving){  
        state = AgentState.DoingNothing;  
        //no action  
        return true;  
    }
```
- 8.)

```
    return false;
```

Actions

```
private void goToRestaurant() {  
    Do("Going to restaurant");  
    host.msgIWantFood(this);  
}  
  
private void SitDown() {  
  
    Do("Being seated. Going to table");  
    customerGui.DoGoToSeat(1);  
}
```

```

private void PickUpAndPeruseMenu() {
    CustomerAgent temp = this;
    timer.schedule(new TimerTask() {
        Object cookies = 1;
        public void run() {
            //look at menu, call waiter when ready
            CallWaiter();
            stateChanged();
        }
    },
    10000);
}

```

```

public void CallWaiter() {
    waiter.msgReadyToOrder(this);
    print("I am ready to order");
}

```

```

private void OrderFood() {

    Random generator = new Random();
    int select = generator.nextInt(4);

    print("Randomed " + select);

    if (select == 0) {
        order = menu.ChooseOne();
    }
    else if (select == 1) {
        order = menu.ChooseTwo();
    }
    else if (select == 2) {
        order = menu.ChooseThree();
    }
    else {
        order = menu.ChooseFour();
    }
    waiter.msgOrderFood(this, this.getOrder());
    stateChanged();
}

```

```

private void EatFood() {

```

```

        Do("Eating Food");
        customerGui.EatTime(order);
        timer.schedule(new TimerTask() {
            Object cookie = 1;
            public void run() {
                print("Done eating, cookie=" + cookie);
                event = AgentEvent.doneEating;
                //isHungry = false;
                stateChanged();
            }
        },
        50000); //getHungerLevel() * 1000); //how long to wait before running task
    }

    private void leaveTable() {
        Do("Leaving.");
        waiter.msgLeavingTable(this);
        customerGui.DoExitRestaurant();
    }

```