**Контрольная точка №2**

**Ресторан «Хачапури и вино»**

Мы решили продолжить дорабатывать логику модели СМО ресторана «Хачапури и вино».

Модель включает в себя четыре страницы: 2D модель ресторана, 3D модель ресторана, диаграммы состояний агентов, а также расчет небольшой статистики по ресторану.

Из задач мы выделили следующие:

1. Определить доли обслуженных и потерянных клиентов.
2. Оценить время обслуживания клиента.

2D модель представлена на рисунке 2 (см. Рисунок 2):

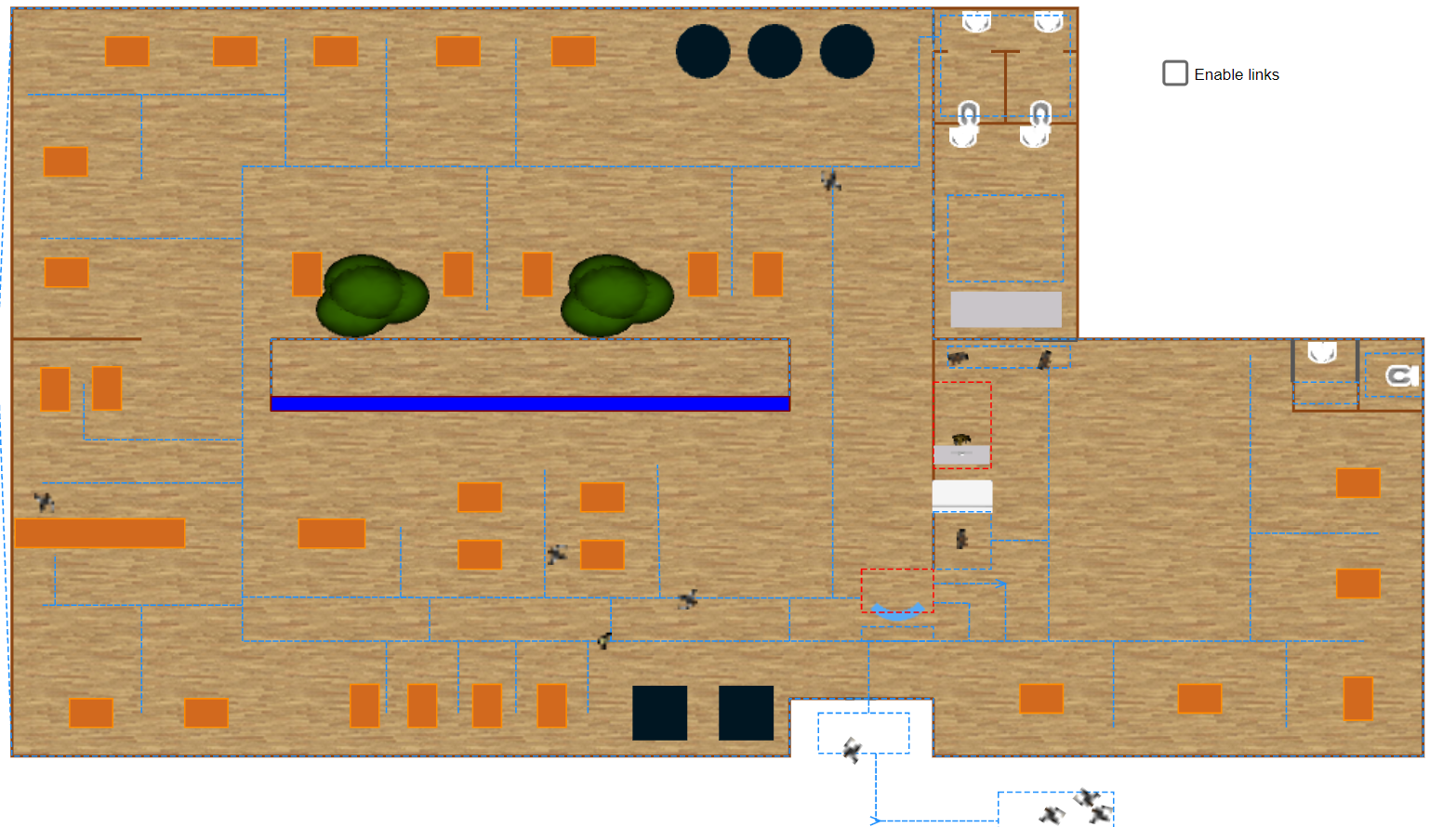


Рисунок 2 - Агентная модель ресторана в 2D

Посетители приходят к входу и ожидают хоста, если ожидание хоста превышает 10 минут, то посетитель покидает ресторан. В нашей модели реализована отдельная территория для тех, кто ушел по таймауту.

После того, как хост встречает клиента, он проводит его к свободному столику. После этого к столику подключается официант и он направляется к своему клиенту, чтобы взять заказ. Как только заказ был принят, официант направляется на кухню. При этом официанты не ограничены своими клиентами. Если у них есть возможность взять заказ у другого клиента, они к нему направляются.

Также реализована возможность для посетителей посещать уборные.

На рисунке 3 (см. Рисунок 3) представлена реализация модели в 3D:

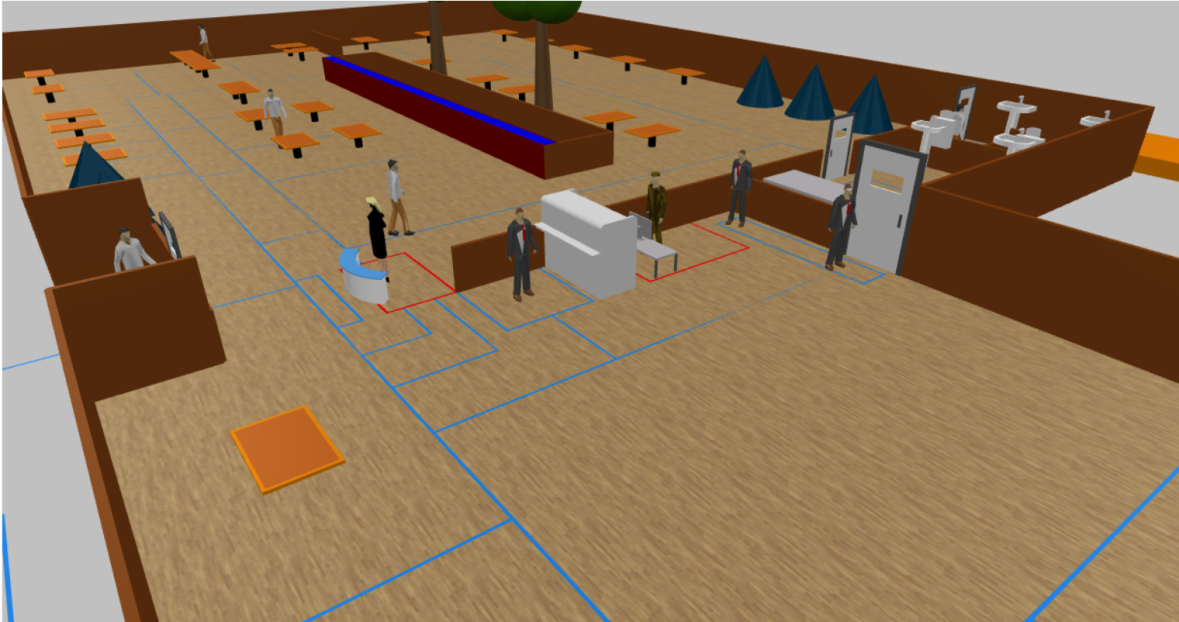


Рисунок 3 - Агентная модель ресторана в 3D реализации

Здесь все происходит так, как было описано выше. Рассмотрим диаграммы состояний всех агентов.

На рисунке 4 (см. Рисунок 4) представлена диаграмма состояний хоста:

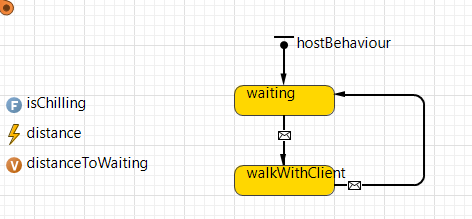


Рисунок 4 - Диаграмма состояний хоста

Хост может либо ожидать у стойки, либо провожать клиента к столику. Действует он в этих состояниях через получение сообщений. Как только клиент приходит в ресторан, хост получает сообщение типа Client и начинает провожать клиента до столика через код moveTo(main.restaurantArea). Как только они доходят до столика, хост получает сообщение от клиента «go» и направляется к своему месту через код moveTo(main.hostNode).

На рисунке 5 (см. Рисунок 5) представлена диаграмма состояний официанта:

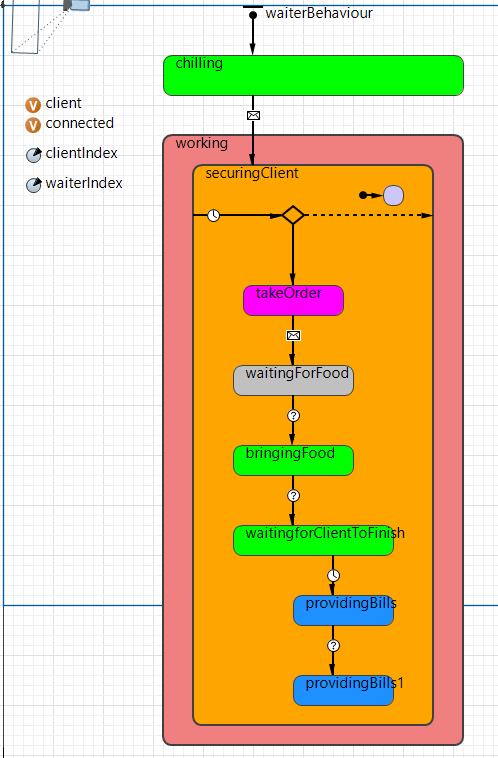


Рисунок 5 - Диаграмма состояний официанта

Официант может либо отдыхать, либо обслуживать клиента. Обслуживание клиента начинается тогда, когда официант получает безусловное сообщение. В этот момент он устанавливает связь с пришедшим посетителем, что представлено на рисунке 6 (см. Рисунок 6):

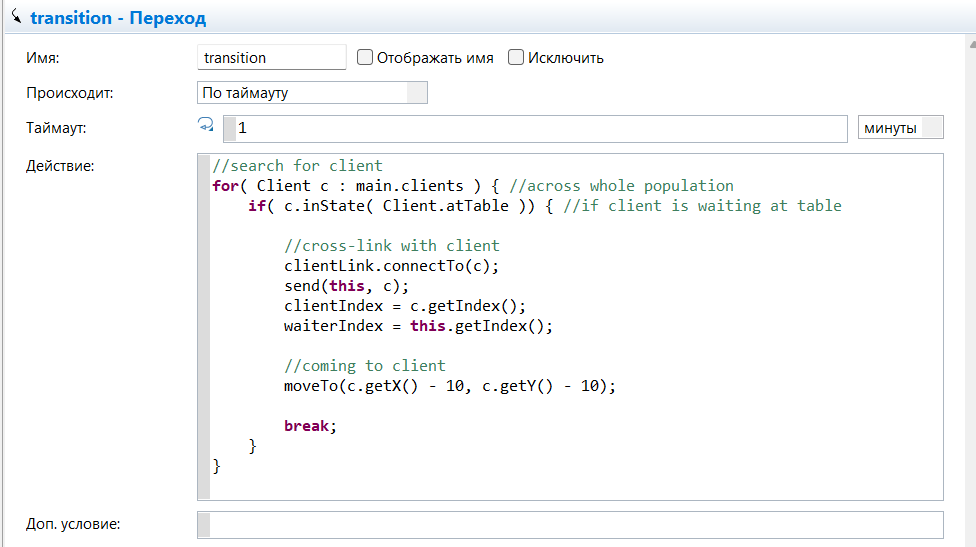


Рисунок 6 - Связь официанта и клиента

После возникновения связи официант направляется к клиенту. Как только он останавливается у столика через код this.isMoving() == false он готов брать заказ и отправляет сообщение клиенту об этом через код send(«ready», main.clients(clientIndex)). Получив от клиента сообщение «yo bring me food -\_-» он направляется к кухне через код moveTo(main.foodNode).

После этого он снова возвращается к клиенту, подает еду, ожидает, как он доест, дает чек, прощается с клиентом и принимает оплату.

На рисунке 7 (см. Рисунок 7) представлена диаграмма состояний клиента:

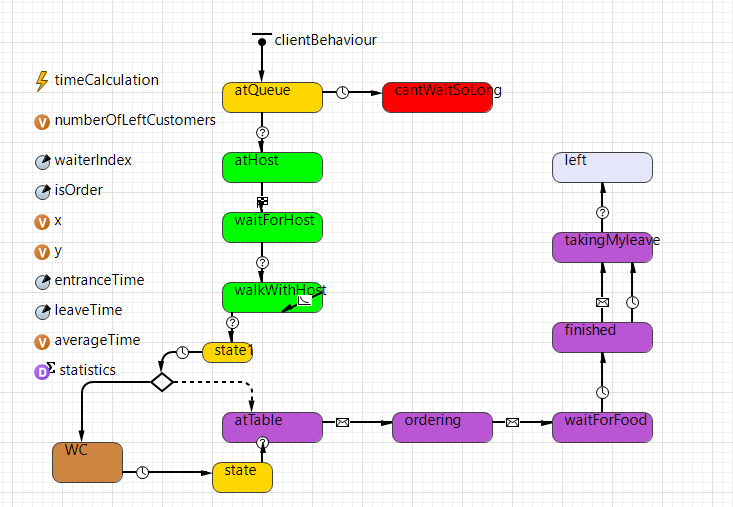


Рисунок 7 - Диаграмма состояний клиента

Как только клиент приходит в ресторан, он отправляет безусловное сообщение официантам, чтобы они начали работать, что представлено на рисунке 8 (см. Рисунок 8):

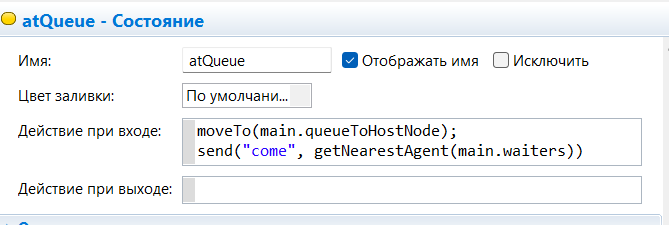


Рисунок 8 - Приход клиента

Как только хост оказывается поблизости через код main.host.isChilling() && main.host.distanceToWaiting < 10, он направляется к стойке: moveTo(main.delayToHostNode). При ожидании более, чем 10 минут, клиент уходит по таймауту. При ожидании хоста он направляет себя как сообщение ему. Как только хост останавливается у клиента main.host.isMoving() == false, они вместе начинают движение к столику через код moveTo( main.host.getX(), main.host.getY()).

Далее клиент отпускает хоста сообщение и может решить, сходить ли в уборную и вернуться к своему столику или сразу начать принимать заказ. Обслуживание официантом происходит через сообщения, как было описано ранее.

После того как клиент пообедал, он направляет сообщение об этом официанту: send(«i finished», main.waiters(waiterIndex)). После этого клиент исчезает из модели.

На рисунке 9 (см. Рисунок 9) представлена статистика по обслуживанию клиентов:

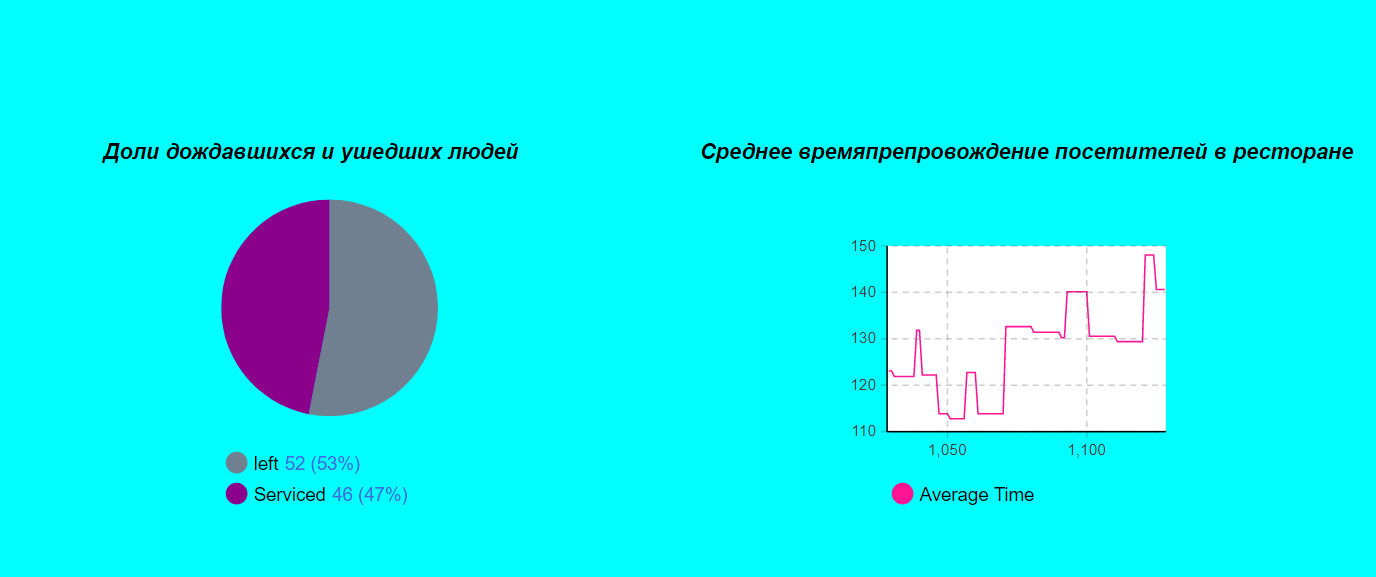


Рисунок 9 - Статистика по агентной модели ресторана

Таким образом, в связи с высокой загруженностью хоста он не успевает обслуживать всех клиентов, поэтому доля потерянных клиентов превышает долю обслуженных клиентов. Здесь стоит отметить, что в реалии «Хачапури и вино» работает на бронировании столиков, поэтому посетители, которые согласовывают свое место с хостом по прибытию, часто вынуждены уйти, так как все столики заняты.

Среднее время обслуживания достигает 1,5 часа и более, что отвечает реалиям ресторана.