

Проект С: Модель павильона метро

Система AnyLogic поддерживает различные подходы моделирования. В данном проекте рассматривается агентный подход к созданию модели наземного павильона метро. Пассажиропотоки и сервисы станции метро моделируются с помощью **Пешеходной библиотеки**. Пешеходная библиотека - высокоуровневая библиотека для моделирования движения пешеходов в физическом пространстве. Она позволяет моделировать здания, в которых движутся пешеходы (станции метро, стадионы, музеи), улицы, парки отдыха и т.д. В моделях, созданных с помощью Пешеходной библиотеки, пешеходы движутся в непрерывном пространстве, реагируя на различные виды препятствий в виде стен и других пешеходов.

Описание проблемы

Пассажиры входят в павильон станции метро через произвольные (случайные) интервалы времени. Внутри павильона расположены турникеты, проверяющие наличие билетов, билетные кассы и платформа отправления. Перед тем, как пройти к поездам метро, пассажиры проходят через турникеты. Те пассажиры, которые не купили билеты заранее, должны будут вначале приобрести их в билетной кассе, и только потом они смогут пройти к поездам. При высокой интенсивности потока пешеходов к билетным кассам образуются очереди, которые препятствуют движению в павильоне. Требуется собрать статистику движения пешеходов, визуализировать пешеходный процесс, построить карты плотности пешеходов в павильоне, вычислить время пребывания пешеходов, выявить проблемы, которые могут возникнуть при перепланировке интерьера здания, и т.д.

Задание С.1. Моделирование потока пассажиров с билетами

Цель: Формализация движения пешеходов, знакомство с пешеходной библиотекой, построение простой пешеходной модели.

1. Провести диалог «Новая модель». Определиться с названием и местоположением. Выбрать единицу модельного времени.
2. Создавать модель пешеходного движения следует с добавления рисунка моделируемого пешеходного пространства. Затем добавляют стены поверх этого рисунка и создают диаграмму пешеходного процесса.
3. Добавить рисунок плана павильона.
 - 3.1. Из палитры **Презентация** перетащите на холст Main элемент **Изображение**. Диалог для выбора файла с изображением появится автоматически. Откройте папку и выберите файл. Также, это можно

сделать в панели **Свойства** для этого элемента. Рекомендуется скопировать изображение плана павильона метро, приведённое на рисунке С1.1.

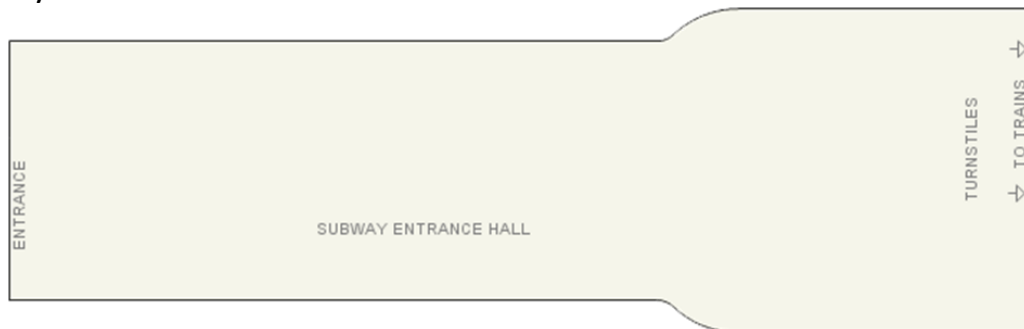


Рисунок С1.1. Изображение плана павильона метро.

- 3.2. . Заблокируйте изображение, установив флажок **Блокировать** в панели **Свойства**. Блокировка нужна, чтобы при рисовании других фигур поверх этого изображения, исключить возможность случайного редактирования изображения.
4. Рисование границ павильона метро.

Граница павильона задаются с помощью объекта **Стена**, играющую роль стен здания. Этот объект находится в разделе **Разметка пространства** палитры **Пешеходная библиотека**. Двойным щелчком мыши по элементу **Стена** активируется режим рисования и можно рисовать стену в графическом редакторе поверх рисунка плана павильона, точка за точкой.

5. Рисование входа и цели движения для потока пешеходов

Линия входа для пешеходного потока задаётся специальным объектом разметки **Целевая линия**. Этот объект находится в разделе **Разметка пространства** палитры **Пешеходная библиотека**. Перетащите объект **Целевая линия** на план павильона в том месте, —ENTRANCE. Объект **Целевая линия** назовите entryLine.

Ещё одну целевую линию разместите на плане павильона в том месте, который обозначен текстом — TO TRAINS. Эта линия играет роль платформы, где пассажиры садятся в поезд и покидают систему. Назовите этот объект — targetLine.

Все элементы разметки (целевые линии и др.) должны находиться внутри стены.

6. Создание диаграммы, задающей поток пешеходов

Рассмотрим простой случай, когда пассажиры входят на станцию метро (целевая линия entryLine) и затем двигаются в направлении платформы к поездам (линия targetLine).

6.1. Диаграмма процесса в AnyLogic создается добавлением объектов библиотеки из палитры на диаграмму типа агентов, соединения их портов и изменения значений свойств объектов.

Добавьте объекты **Пешеходной библиотеки** на диаграмму и соедините их так, как показано на рисунке С1.2.

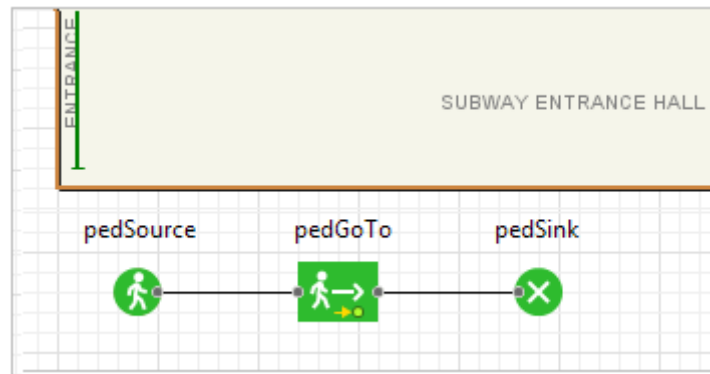
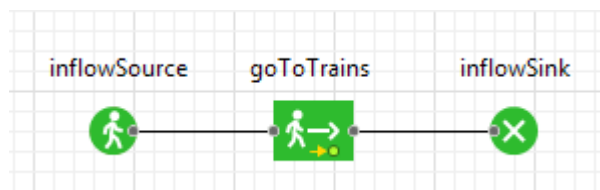


Рисунок С1.2. Диаграмма простого процесса.

Переименуйте объекты. Назовите их *inflowSource*, *goToTrains*, *inflowSink*.



Кратко опишите суть и свойства объектов диаграммы:

Объект [PedSource](#) создает пешеходов. Обычно он используется в качестве начальной точки диаграммы процесса, формализующей поток пешеходов. В нашем примере он моделирует приход пассажиров в павильон.

Объект [PedGoTo](#) моделирует перемещение пешеходов из текущего местоположения в другое (заданное параметром этого объекта). С помощью этого объекта мы будем моделировать то, как пассажиры перемещаются от входа в павильон к поездам метро.

Объект [PedSink](#) удаляет поступивших в объект пешеходов из моделируемой среды. Обычно объект используется в качестве конечной точки диаграммы процесса.

6.2. Уточнение свойств объектов диаграммы

Выделите объект *inflowSource*. В панели **Свойства** задайте место, где появляются пассажиры. Выберите *entryLine* (название нашей целевой линии, нарисованной ранее у входа) из выпадающего списка **Целевая линия**.

Задайте 4000 в час в параметре **Интенсивность**.

Выделите объект *goToTrains*. После того, как пассажиры войдут в здание, они будут двигаться к той цели, которую вы здесь укажете. В данном случае, укажите *targetLine* в списке **Целевая линия**.

Оставьте все свойства объекта **PedSink** установленными по умолчанию.

7. Эксперимент этой модели назовите Subway Entrance Hall/Simulation.

Проведите эксперимент, наблюдаем моделируемый процесс, видим, как пассажиры входят в павильон и движутся по коридору, ведущему к поездкам метро.

Исследуйте зависимость пропускной способности павильона метро за неделю в зависимости от интенсивности входа.

8. Подготовьте отчёт с выводами, продемонстрируйте работу модели и ответьте на вопросы преподавателя.

Задание С.2. Моделирование турникетов и автоматов продажи билетов

Цель: Освоить технологию применения сервисов в пешеходных моделях.

1. Моделирование сервисов

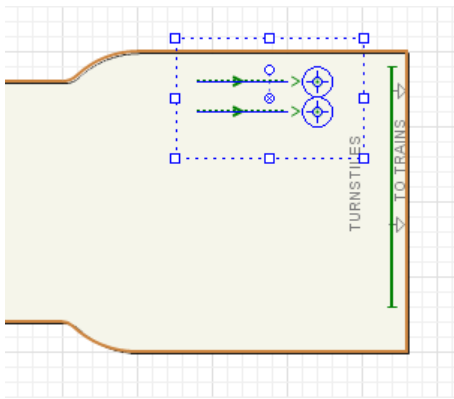
Добавим в модель турникеты для проверки билетов. Турникеты являются, типичным примером использования сервисов в моделях с пешеходами. Есть два типа элементов разметки пространства, которыми можно воспользоваться, чтобы добавить сервисы в модель:

Сервис с очередями - используется для того, чтобы задавать сервисы, в которых пешеходы ждут в очереди, пока сервис не будет доступен.

Сервис с областью - используется для того, чтобы задавать сервисы с электронной очередью. В таком случае пешеходы не стоят в очереди, а ждут в расположенной рядом области.

Турникеты обычно моделируются элементом **Сервис с очередями**.

Нарисуйте турникеты. Для этого перетащите элемент **Сервис с очередями** из раздела **Разметка** палитры **Пешеходная Библиотека** в графический редактор. Увидим две точки сервиса и две очереди, ведущие к этим точкам. Разместите эти элементы, как показано на рисунке:



Настройте сервисы. Назовите их **fareGates**. Увеличьте **Количество сервисов** и очередей до 6. Измените значение свойства **Тип сервиса** с **Точечный** на **Линейный**.

2. Изменение диаграммы процесса

Перетащите объект **PedService** из палитры Пешеходной библиотеки в диаграмму модели перед объектом *goToTrains*. В окне **Свойства** измените название на *checkTickets*. Свяжите объект с элементом разметки **fareGates**. Оставьте без изменения значения поля **Время задержки**.

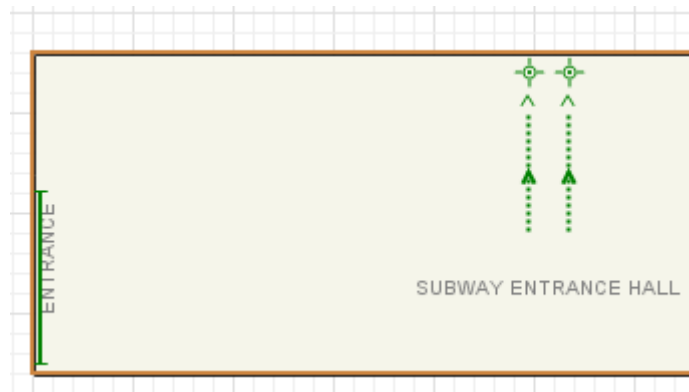
3. Эксперимент с турникетами

Запустите модель. Наблюдайте процесс образования очередей перед турникетами. Увеличьте количество турникетов. Так чтобы очередей не было. Данная модель помогла нам найти требуемое количество точек сервиса. Вы можете изменять интенсивность пассажиропотока в настройках блока **RedSource** и наилучшим образом моделировать средства обслуживания согласно нагрузке.

4. Добавление автоматов продажи билетов

Не все люди входят в павильон метро с билетами в кармане, многие покупают билеты, лишь, когда заходят в станцию метро. Добавим в нашу модель автоматы продажи билетов. С помощью созданной модели можно будет определить необходимое количество автоматов, чтобы успешно обслужить всех пассажиров и найти оптимальное место расположения автоматов, чтобы минимизировать пересечения потоков пассажиров и образование пешеходных пробок.

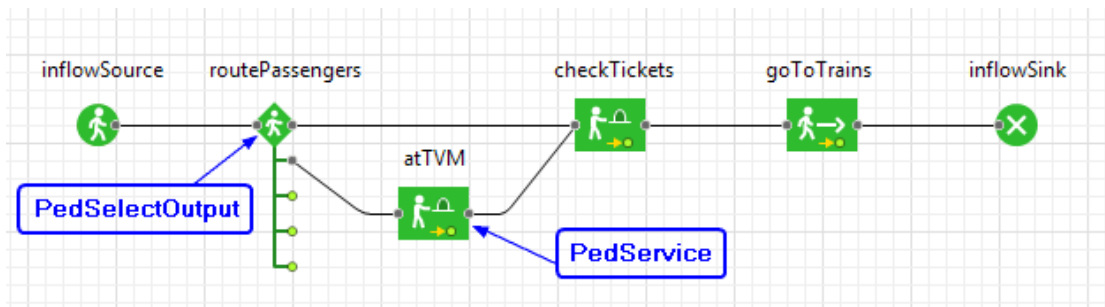
Автоматы продажи билетов логично моделировать элементом **Сервис с очередями**. Перетащите элемент **Сервис с очередями** из секции **Разметка** палитры **Пешеходная Библиотека** в графический редактор. Разместите их так, как показано на рисунке:



Настройте свойства сервиса. Для этого оставьте выбор **Точечный** в свойстве **Тип сервиса**. Назовите сервисы *ticketMachines*. Увеличьте параметр **Количество сервисов** до 7. Соответственно, увеличьте параметр **Количество очередей** так же до 7.

5. Разделение потоков пассажиров

- 5.1. Следует внести изменения в диаграмму модели, для того, чтобы направить пассажиров с билетами к турникетам, а пассажиров с без билетов к билетным автоматам. Объект **PedSelectOutput** является блоком принятия решения Пешеходной библиотеки. Пешеход, вошедший в блок **PedSelectOutput**, будет перенаправляться в один из пяти выходных портов в зависимости от заданных для этих портов коэффициентов предпочтения.
- 5.2. Измените свойства объекта **PedSelectOutput**. Назовите его *routePassengers*. Укажите значение 0.7 в поле Коэфф. предпочтения 1 (коэффициент для потока, направляющегося напрямую к турникетам) и значение 0.3 в поле Коэфф. предпочтения 2 (коэффициент для потока пассажиров, направляющихся к автоматам продажи билетов соответственно). Укажите в полях Коэфф. предпочтения 3, 4, 5 значение, равное 0.
- 5.3. Добавьте ещё один объект **PedService**. Этот блок будет моделировать обслуживание пассажиров у автоматов продажи билетов. Поместите его между объектом *PedSelectOutput* и ранее созданным объектом *PedService* (*checkTickets*). Переименуйте объект *PedService*. в *atTVM*. Выберите *ticketMachines* (название элемента разметки Сервис с очередями) в свойстве Сервисы. Измените параметр Время задержки. Введите в поле: *triangular(7, 12, 40)* и выберите секунды в качестве единиц времени, т.е. время обслуживания неравнозначно распределено с минимальным значением 7 секунд, средним 12, и максимальным 40 секунд.
- 5.4. Соедините блоки, как показано на рисунке.



Запустите модель и понаблюдайте за её динамикой. Теперь некоторые пассажиры перед тем, как пройти к турникетам, сначала подходят к кассам, чтобы приобрести билет.

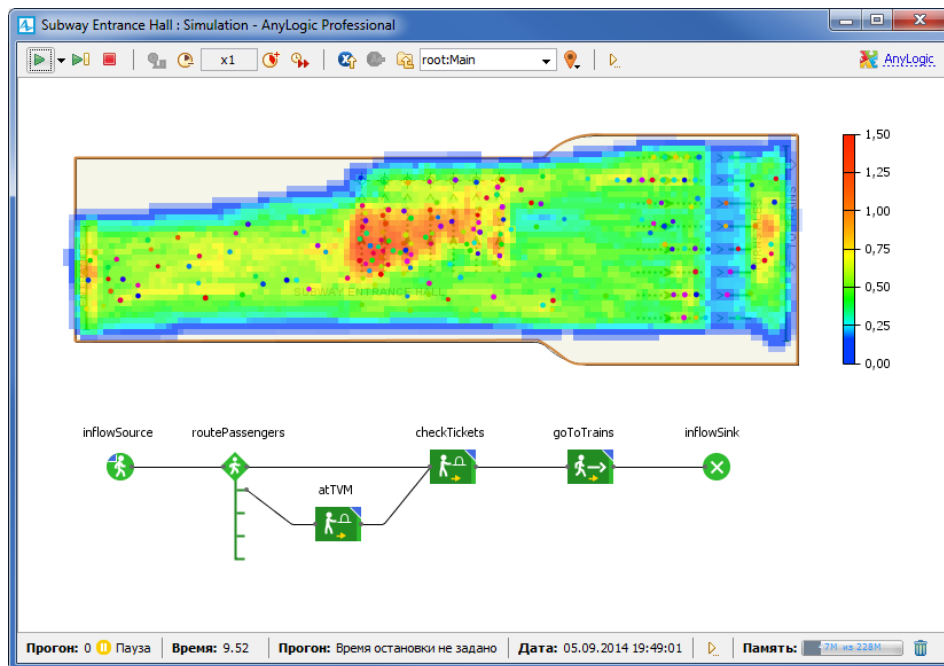
Самостоятельно добавьте два эскалатора

6. Карты плотности пешеходов

AnyLogic поддерживает сбор статистики по плотности пешеходов в моделируемом пространстве и отображение этой информации на презентации в виде **карты плотности**.

6.1. Для того, чтобы отобразить карту плотности пешеходов перетащите элемент **Карта плотности пешеходов** из секции **Разметка пространства** палитры **Пешеходная библиотека** в графический редактор. На анимации отображается легенда карты. Легенда карты плотности помогает понять, какие цвета соответствуют каким значениям плотности. Откройте панель **Свойства** этого блока и сбросьте флажок **Отображать имя**. Выберите значение *ground* из списка **Этаж**.

6.2. Запустите модель и наблюдайте за динамикой моделируемого процесса.



По мере того, как пешеходы двигаются в моделируемом пространстве, план помещений будет закрашиваться различными цветами. В каждой точке пространства цвет будет соответствовать измеренной в этой точке плотности пешеходов. Карта плотности постоянно перерисовывается в соответствии с актуальными значениями плотности.

Карта плотности чаще всего используется для обнаружения участков пространства, на которых значение плотности достигает критических значений. Такие области отображаются на карте плотности красным цветом. По умолчанию значение критической плотности задано равным 1,5 пешехода на квадратный метр. Это значение можно изменить в свойстве **Критическая плотность (отображается красным), пешеходов/м²** элемента **Карта плотности пешеходов**. При нулевой плотности закрашивание соответствующего участка не производится вообще

По умолчанию AnyLogic использует логарифмическую цветовую схему. При логарифмической схеме цвет стремительно приближается к "критическому" (красному) только при приближении к зоне критических значений плотности, а при малых значениях остаётся нейтральным. Можно сменить логарифмическую схему на линейную, выбрав **Линейная** в свойстве **Цветовая схема**. В этом случае цвета будут меняться от синего к красному линейно согласно градиенту спектра цветов. При желании можно задать и собственную цветовую схему любой сложности, выбрав **Другая** в свойстве **Цветовая схема**. Так же можно выбрать другие цвета.

Можно отключить отображение карты плотности, но при этом продолжать собирать соответствующую статистику, без отображения карты при анимации модели. Для этого сбросьте флажок **Показывать карту плотности на анимации**. В таком случае Вы можете увеличить скорость выполнения мо-

дели. Эта статистика может храниться в наборах данных, откуда, например, по окончании моделирования, записываться в базу данных для последующего статистического анализа.

Контрольные вопросы к проекту С.

1. В чем различие понятий *Линейные* сервисы и *Точечные* сервисы?