Pedestrian Dynamics Tutorial

Pedestrian Dynamics 3

Written by: INCONTROL Simulation Solutions

Papendorpseweg 77

3528 BJ Utrecht

The Netherlands

www.incontrolsim.com

Date: 6-3-2019 10:53

Version: 3

1 Table of Contents

2 Getting Familiar with Pedestrian Dynamics

2.1 Pedestrian Dynamics

Pedestrian Dynamics is a software package to simulate pedestrian flows in and around buildings. It can be used to optimize the infrastructure of train stations, airport terminals, museums, shopping areas, sports venues and stadiums, hospitals, exhibition halls etc. With Pedestrian Dynamics you can easily build simulation models and show results about quality and safety of an infrastructure, by evaluating normal operation and performing evacuation scenario's. You can quickly build a 3D model of buildings and their walking areas. The behavior of the pedestrians is determined by parameters such as a minimum and maximum walking speed but most importantly by the list of activities they have to perform within the environment. Activities can be buying a ticket or waiting at certain locations in the area. It is easy to create groups of pedestrians, each with different walking speeds, 3D representations and destination lists.

Having created such a model, an experiment can be run to simulate the pedestrian flow through the building. After the run it is easy to examine the output. You can perform all kinds of measurements concerning the walking times and pedestrian densities that occur in the building.

In this tutorial you will learn the basics of working with Pedestrian Dynamics. You will be introduced to the elementary building blocks of the software and how to use them. You will also learn how to run an experiment and how to measure the results of the experiment.

2.2 Starting Pedestrian Dynamics

Pedestrian Dynamics can be started using the start menu. During this phase, a splash screen, see Figure 1 first appears.

Figure 1: Splash screen of Pedestrian Dynamics

As soon as Pedestrian Dynamics is started up, you see several windows: a Main menu bar, a Start Page, and a Model Layout, see Figure 2 and 3 for an overview.

Main menu bar	This is used, among others, for opening and saving files.
Start Page	An overview of resources available to get you started.
Model Layout	This is where the model is developed.

Figure 2: Description table of the windows visible after starting Pedestrian Dynamics.

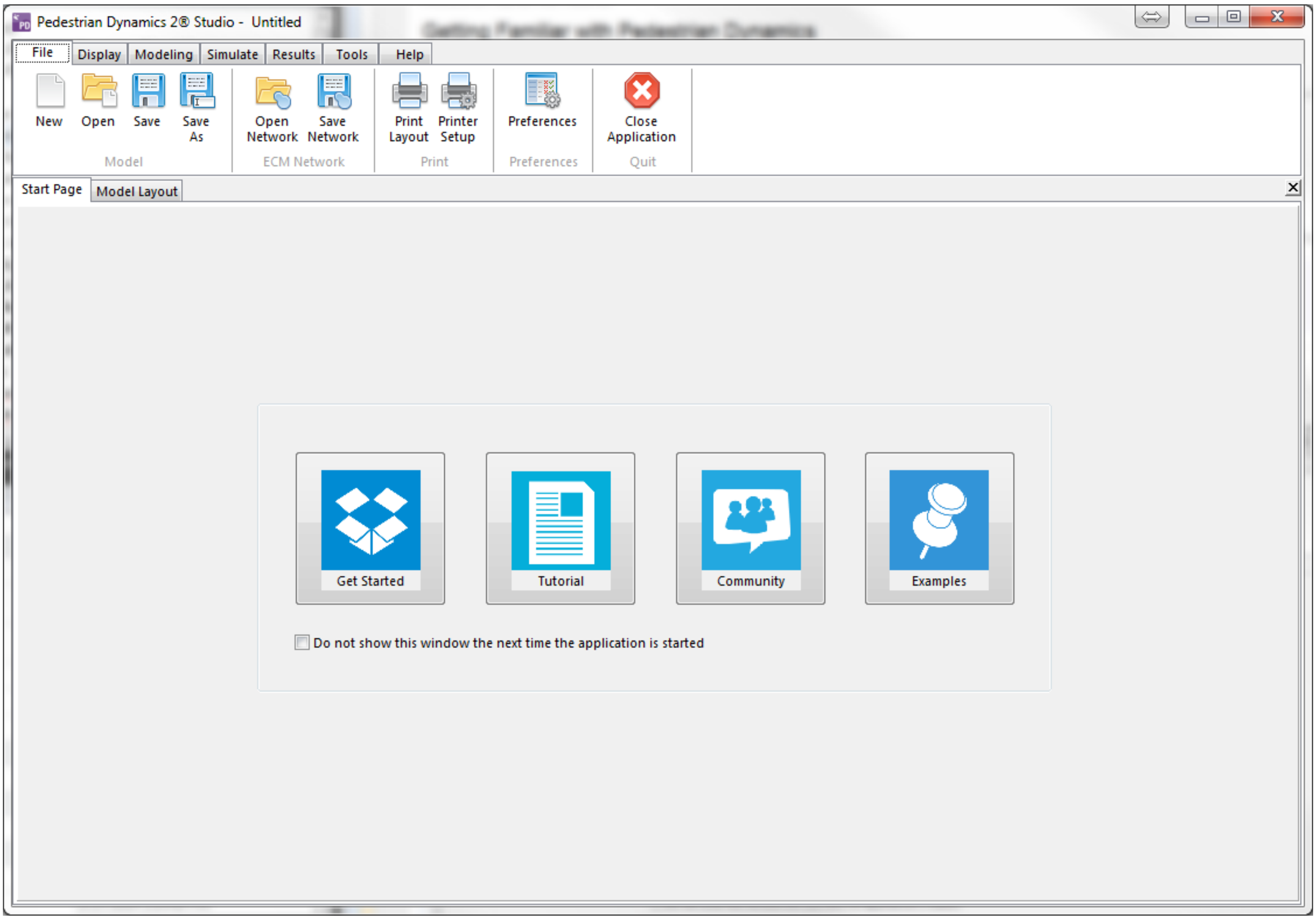


Figure 3: Layout of the opening window of Pedestrian Dynamics.

The function and the appearance of the menus are similar to those in other Windows applications, such as Word and Excel. The most used menu options are explained in the table below.

The main menu bar is subdivided as follows:

File	Make, open or save files, to set preferences and to control standard functions such as printing.
Display	Open viewer and layout windows.
Modeling	Model input settings and network creation.
Simulate	Run controls to run a simulation model. Design and perform an experiment.
Results	To generate reports and graphics of a *single* simulation run or evaluate results of an experiment.
Tools	Contains tools such as the 4DScript interact and Autofit to fit a distribution to given data.
Help	Open the documentation, the Example Wizard as well as to find company and version information.
Developer	Tools useful for developer to create Libraries an attribute function.

Each main menu item is divided in to groups. See the documentation for a complete description.

3 Building a pedestrian flow model

In this chapter we will start with building some simple pedestrian flow models in Pedestrian Dynamics.

The objective is to get familiar with the concepts Agent, Height Layer, Obstacle, Activity, Activity Locations and Activity Route and to learn how to build a simple model in Pedestrian Dynamics, i.e. drawing the environment and defining the behavior of the pedestrians within the environment.

First, a quick overview of how to build and simulate a pedestrian flow model.

When building a pedestrian flow model, you model both the physical environment and the behavior of the pedestrians in this environment. The environment includes the layout of floors and walls, stairs and escalators, designated walking areas, etc. The pedestrians are modeled by Agents. The behavior of an Agent is determined by various parameters such as walking speed and most importantly by the list of Activities the Agent plans to perform within the environment. An Activity could be buying a ticket which can take place in a ticket facility or buying groceries in a shop. A Ticket facility and a Commercial facility (shop) are examples of an Activity Location. These are areas where an Activity can take place. When we design the environment, we also have to create these Activity Locations in our model.

When finished building the model, you can run an Experiment. Based on the Agent input settings, Agents will be created in the environment. Each agent will decide on a path through the environment so that it can fulfill its planned Activities at the various Activity Locations. After running the experiment the results are saved and can be analyzed at any time: you can examine the crowdedness of the whole environment or get specific data on corridors, stairs or any other area.

We will first focus on modeling the environment.

> The dimensions don’t really matter in the tutorial model. It is more about the techniques you learn. However, if you want to create an exact replica then the easiest way is to open an additional PD application in which you load the supplied tutorial model. You can then obtain the exact dimension from here.

3.1 Drawing the environment

The first step in building a model in Pedestrian Dynamics is to draw the environment. When you draw the environment you always start with a Height layer. The Height layer bounds the area where the Agents can walk. When you create a new model in Pedestrian Dynamics it already contains an initial Height layer, see Figure 1. This layer is typically used to represent the first floor of a building, but it can also include the surroundings of the building. If the environment that is modeled has several floors, more Height layers must be added, one for each floor.

Within the Height layer you draw Obstacles to create the physical environment. Obstacles are areas where Agents are not allowed to walk. These could be walls, statues, fountains or any other objects that Agents have to avoid when planning their route through the environment.

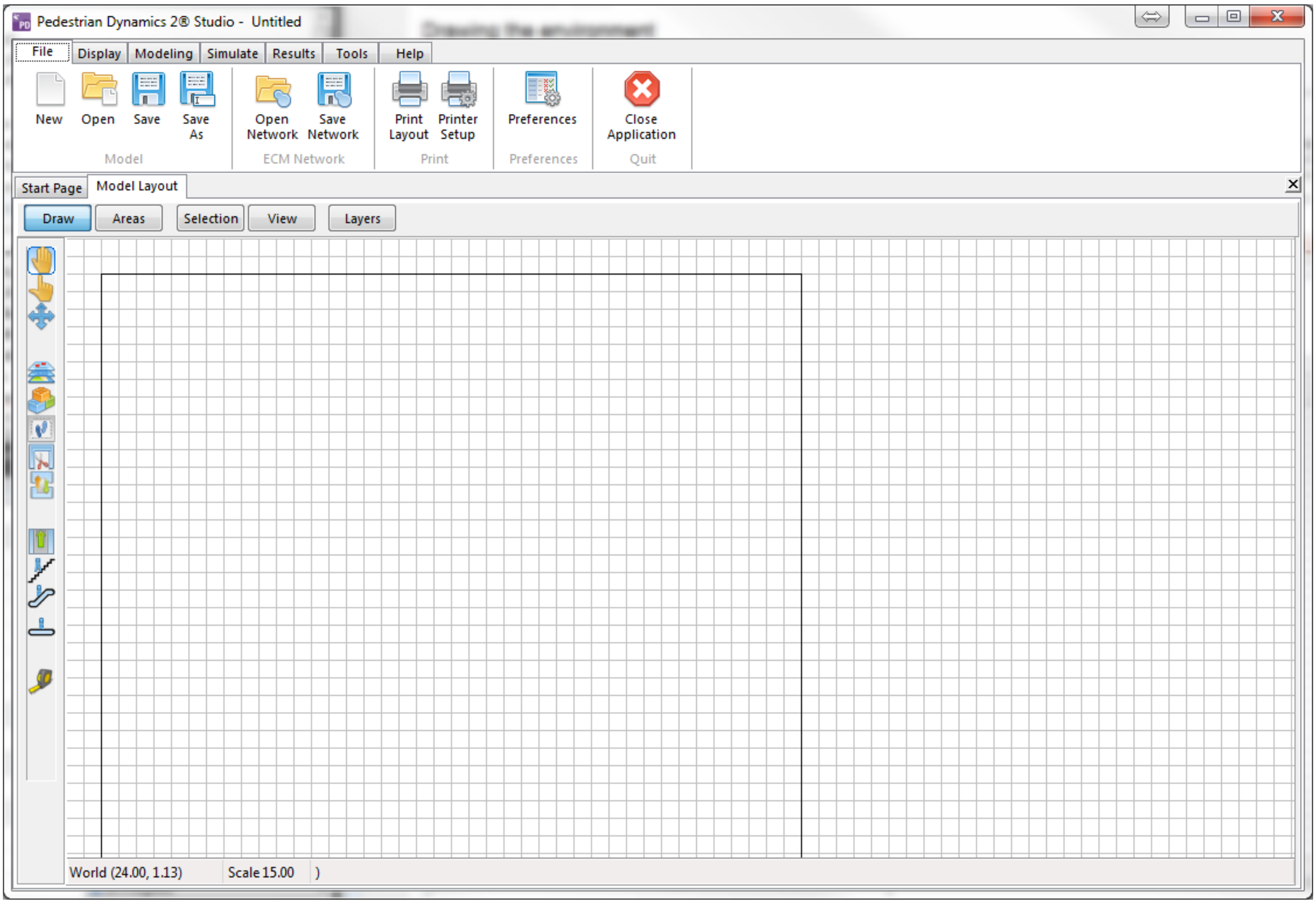


Figure 1: The Model Layout window with one Height layer. The black square is the outline of the layer. On the left, the drawing toolbar is visible.

Case 1: The pedestrian tunnel

A pedestrian tunnel under the railway connects the town center with residential areas in the West part of town. The tunnel is also used as a train station. The tunnel is at street level. In the center of the tunnel, a flight of stairs and an escalator lead to the platforms. The tunnel contains two concrete columns to support the tracks above.

A growing number of people use the tunnel either to go to the platforms or just to go from the town center to the West or vice versa. A second flight of stairs or escalator may be necessary to deal with the growing number of pedestrians.

Exercise 1

Create a model of the tunnel with pedestrians entering and exiting the tunnel on the town center side and West side. The pedestrians have to avoid the columns to pass through the tunnel.

We will first draw the environment. We use a Height layer to model the floor of the tunnel and an Obstacle for each of the concrete columns.

The Height layer that represents the tunnel floor has already been drawn in the Model Layout. Resize the Layer to get a tunnel shape. Select the second 'pointing hand' icon from the top in the toolbar on the left hand side of the Model Layout, see Figure 1, to be able to select and resize the Layer. If no toolbar is visible, click on Draw in the menu bar of the Model Layout. The next step is to select the Layer by clicking it with the mouse. Black squares will appear on the sides and corner of the Layer. To resize the Layer, drag one of the small black squares on the edge of the Layer to the desired size, see Figure 2.

When you resize an element such as a Layer, the edges of the element will align with the grid lines. This is caused by the *Snap to grid* option. By default *Snap to grid* is enabled and the lines of the grid are separated by 0.1 meter, this is the *Grid size*. Use the buttons in the View menu of the Layout window to hide the grid, set the *Grid size*, and disable the *Snap to grid*. For this case we will use a grid size of 0.5 meter. Switch to the View menu and set the grid size to 0.5 meter.

When drawing the environment the *Snap to grid* option is often helpful. When you resize an element such as a Layer, the edges of the element will align with the grid lines if *Snap to grid* is enabled. Note that you can open several Layout windows and give them each different Snap to grid settings. Open for example two Model Layout windows and give them a different *Grid size*.

A general *Grid size* can be set from the Modeling page of the General settings dialog that can be opened from the Modeling tab of the main menu. Each new layout window that is opened will have this general *Grid size* as default value.

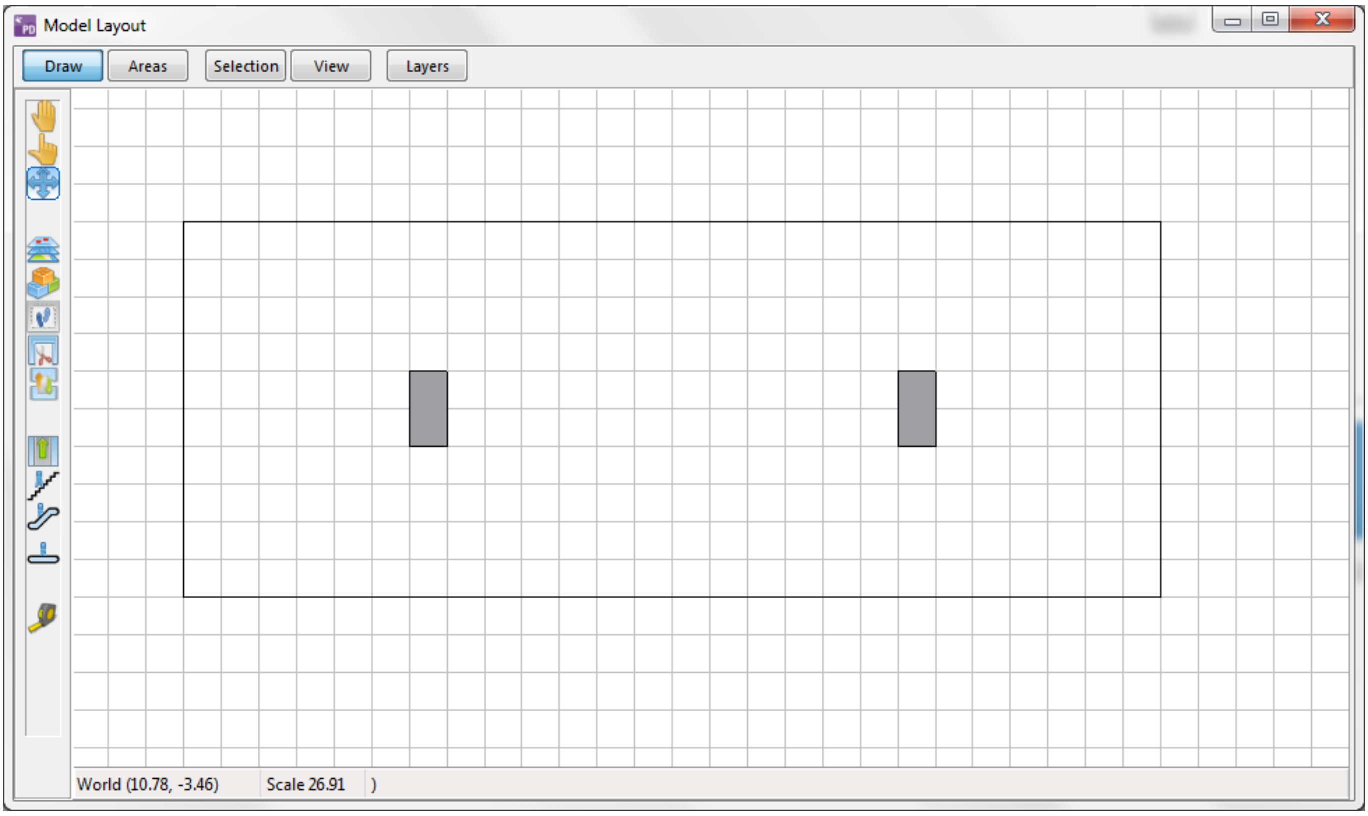


Figure 2: The Model Layout window showing the tunnel with the two Obstacles that represent the concrete columns that support the tracks.

The next step is to draw the concrete columns, which will be represented by Obstacles. Both Height layers and Obstacles are drawn in the same way. Make sure that the drawing toolbar is visible on the left of the Model Layout, see Figure 2. If it is not visible click 'Draw' in the menu bar of the Model Layout. Note that the fourth icon from the top represents a Height layer and the fifth an Obstacle. Click on the Obstacle icon. A second toolbar with a collection of shapes will appear. In this toolbar, the shape of the Obstacle can be selected. When drawing an Obstacle, several ready-made shapes such as a square, rectangle, circle or line are available, but a custom shape can be created using a polygon. For this exercise, select the rectangle. Move the mouse to the Height layer and click the left mouse button while holding the Ctrl-key. One of the corners of the rectangle is now fixed. If you move the mouse, the rectangle changes shape. Use Ctrl-click again to finalize the shape of the Obstacle.

Do the same to draw the second column, or make a copy of the first one. To make a copy, first switch to the 'Select' mode, the second 'pointing hand' icon in the Drawing toolbar. Then select the first column and press Ctrl-C to copy and Ctrl-V to paste. The new copy has been created but at the same position as the first. To move the copy, switch to the 'Move' mode by selecting the third icon, the icon of the arrow pointing in four directions. Drag the second column away from the first and position it somewhere to the right. Alternatively, the position of an element can also be set using exact coordinates. Switch to the 'Select' mode and double click the column, a properties dialog will appear. On the coordinates page you can set the x, y position of the Top Left and Bottom Right corner.

3.2 Entering and exiting

The next step is to add pedestrians to the model. Pedestrians (Agents) enter the model by performing an 'Entering' Activity and leave the model by performing an 'Exiting' Activity. In Pedestrian Dynamics, every Activity always takes place in an Activity Location. There are several types of Activity Locations. Technically each of these types is the same and can be used in any situation. Typically we will use an Entry/Exit area as a location in our environment from which Agents can enter and leave the model. We will now add Entry/Exit areas for the East and West entrance of the tunnel. To draw an Activity Location click on ‘Areas’ in the menu of the Model Layout. The Activity Toolbar will appear on the left hand side of the Model Layout window. Notice that the first three icons, the Pan, Select and Move modes, are identical to those in the Draw Toolbar. Click on the 'green arrow pointing to an open door' icon to draw an Entry/Exit area. A second tool bar with a collection of shapes will appear. Select the rectangle and move the mouse to the Height layer. Again use the Ctrl-key to fix the top left and bottom right corner. In the same way, add a second Entry Exit area, see Figure 3.

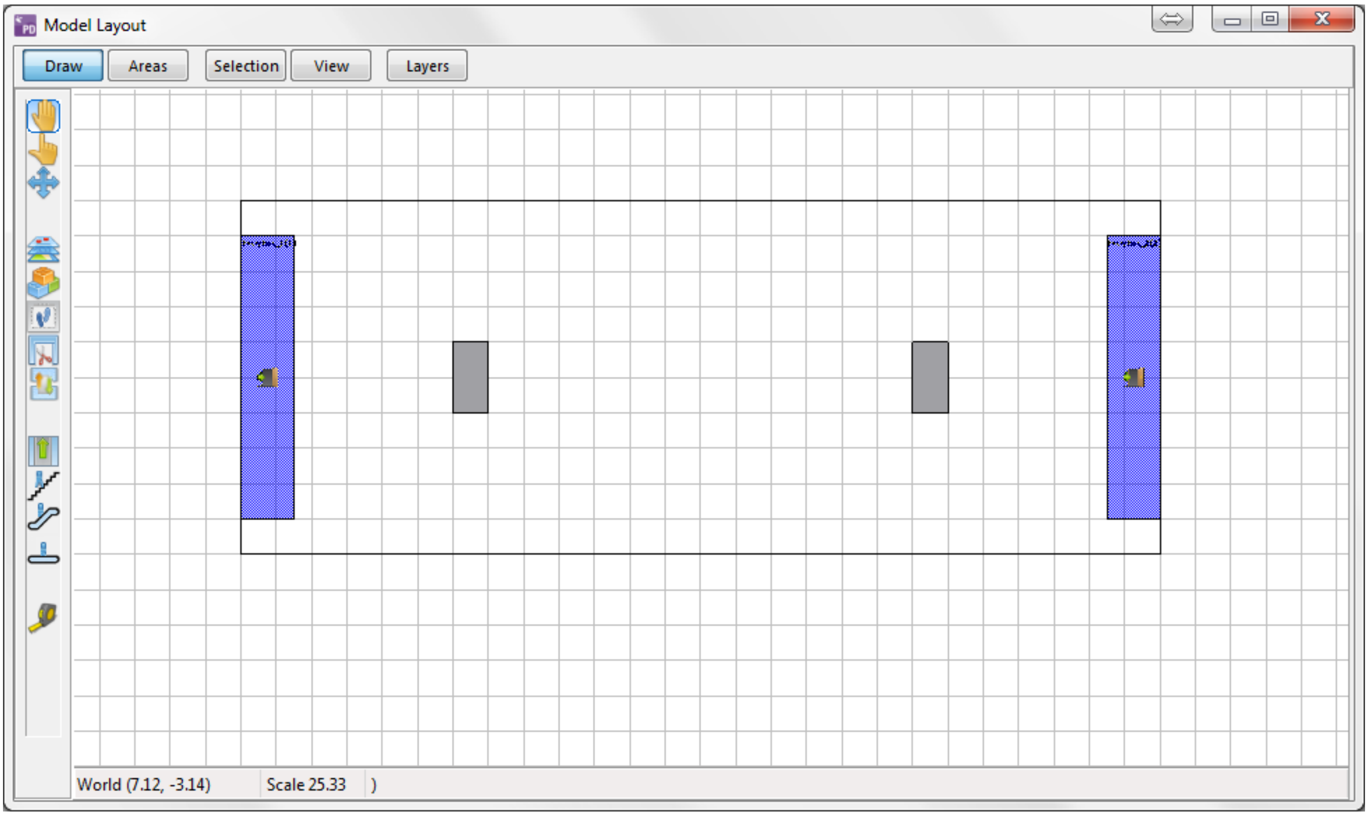


Figure 3: We are now ready to run our pedestrian flow model for the first time. With the default settings, pedestrians will enter through either of the Entry/Exit areas and leave the model through the other.

To start the simulation, press the Run button of the Run Control, see Figure 4. You can find the Run Control in the Simulate tab of the main menu. A network of purple lines will appear. This is the Explicit Corridor Map which is used by Agents to determine their path through the environment avoiding all obstacles. During the simulation run you will see that small blue dots are created at the Entry/Exit areas. These dots represent the pedestrians. Watch how the pedestrians make their way through the environment without running into the obstacles. Adjust the horizontal slide bar in the Run Control to change the speed at which the simulation runs.

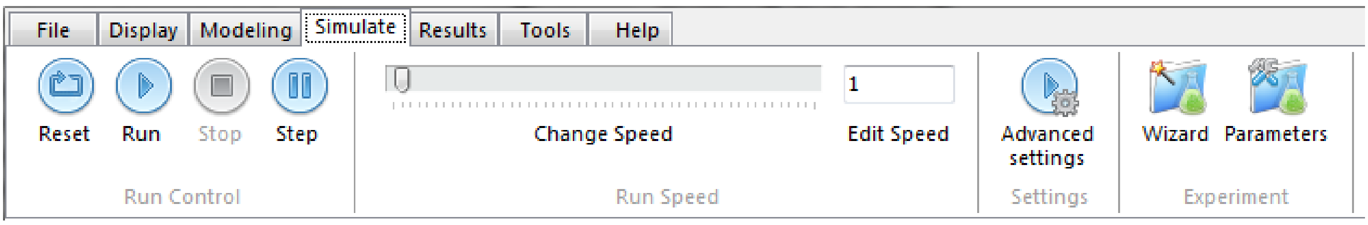


Figure 4: The run controller page in the main menu

3.3 Activity Locations

The tunnel is not only used as a passageway and a train station, it also contains two shops on the North side of the tunnel. Pedestrians that use the tunnel and do not plan to visit the shops have to avoid them, they cannot walk through them. Only pedestrians that visit the shops will go there. We use Obstacles to allocate the area for the shops. This ensures that pedestrians who only use the tunnel to go to the other side will plan their route around the shops instead of walking right through them. At the entrance of the shop, we create an Activity Location to allow customers to visit it.

During their stay in the model, Agents perform Activities. An Activity can be buying a ticket, shopping, waiting in a waiting area or exiting the building. Activities can often take place in several locations. For instance, a building can have several exits and a railway station can have more than one ticket facility, these are the Activity Locations. There are several types of Activity Locations available in Pedestrian Dynamics. We have already used the Entry/Exit area where Agents are created and leave the system. A second example of an Activity Location is a Commercial facility. A Commercial facility represents a shop. When an Agent visits the shop, a property on the Commercial facility will determine the probability distribution of the time that the Agent spends in the shop.

Technically there is no difference between the different types of Activity Locations. The different types make it easy to create groups of locations where a certain kind of activity can be performed by the agents.

Exercise 2

> Create a gift shop and a flower shop on the North side of the tunnel.

The first step is to reshape the tunnel to create more space on the North side of the tunnel to accommodate the shops. Do not forget to switch to the 'Select' mode. Press the "s" key to quickly switch to the select mode. The second step is to allocate space for the shops by creating Obstacles. Finally, switch to the Activities toolbar, click the Commercial facility and select a shape. Draw rectangular Commercial facility areas at the entrance of the two shops, see Figure 5. Run the model and see what happens.

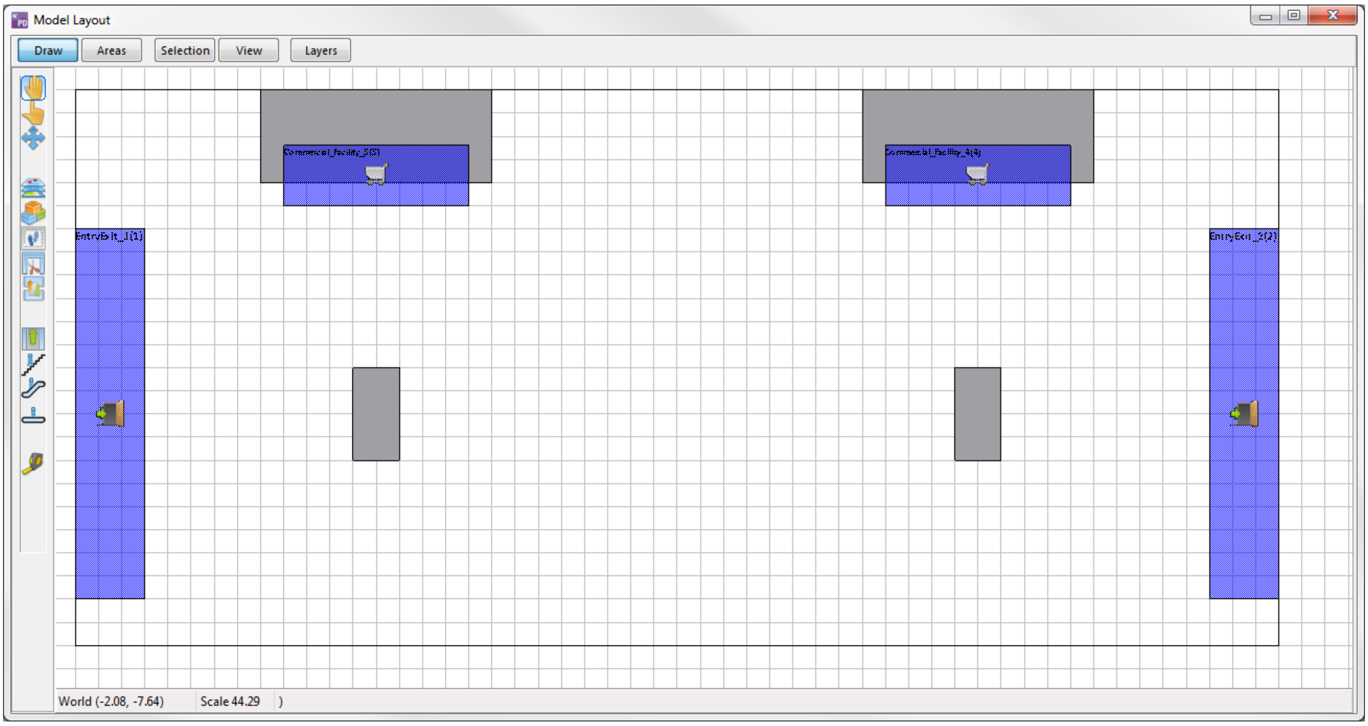


Figure 5: The tunnel with two Commercial facility positioned at the North side.

During the simulation run, we see the pedestrians move through the tunnel while they avoid both the columns and the shops. They do not yet visit the shops.

3.4 How agents plan their way through the environment

With the current settings, the Agents do not visit the shops. They are created in one of the Entry/Exit areas and then walk through the tunnel avoiding the obstacles and leave the model through the other Entry/Exit area. This is the default behavior. This behavior can be modified but therefore we need to understand how it is created. The behavior of the Agents is determined by various parameters, but most importantly by the list of Activities the Agent plans to perform within the environment, its Activity Route. Recall that an Activity can be buying a ticket, shopping, waiting in a waiting area or exiting the building. The first Activity of an Agent is almost always to enter the environment through an Entry/Exit area. Subsequently, the agent will follow a list of desired Activities, its Activity Route. We will now take a look at the current Activities and the Activity Route. On the Modeling tab of the main menu click Agent input, the Agent Input Settings dialog will appear, see Figure 6.

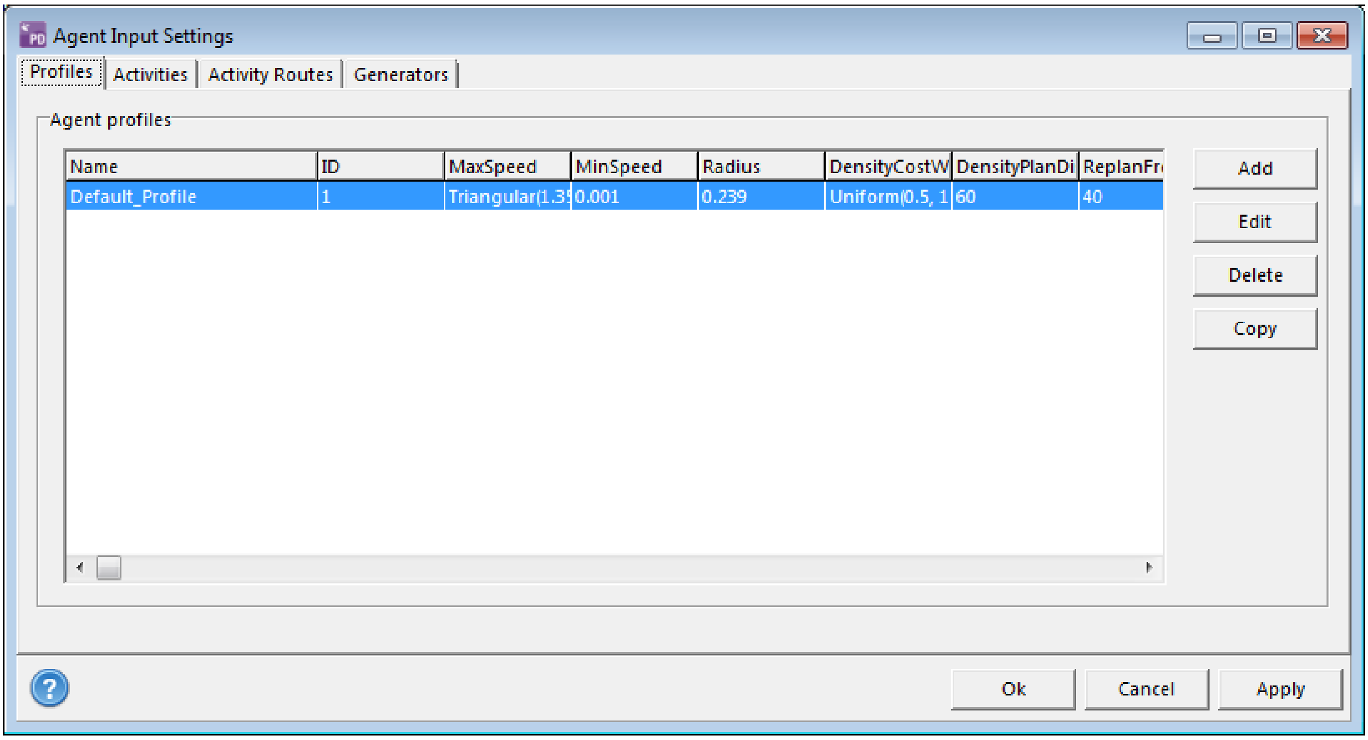


Figure 6: The Agent Input Settings dialog.

Switch to the Activities page. Note that there are currently two activities defined: one Entry and one Exit Activity, see Figure 7.

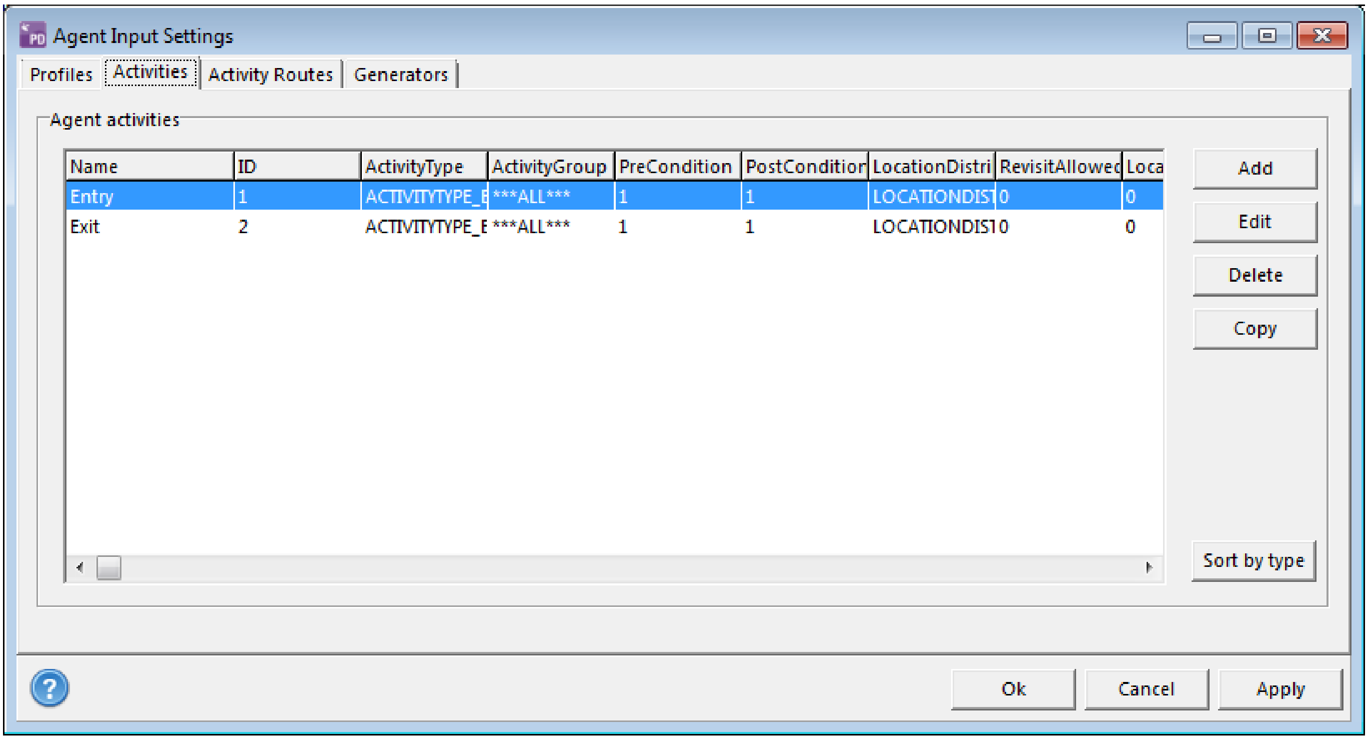


Figure 7: The Activity page of the Agent Input Settings dialog, with two Activities.

If we switch to the Activity Routes page, we see one route named Default_Route, see Figure 8. In this route, two activities are to be performed: the Entry Activity and then the Exit Activity. Recall that these activities were defined on the Activities page.

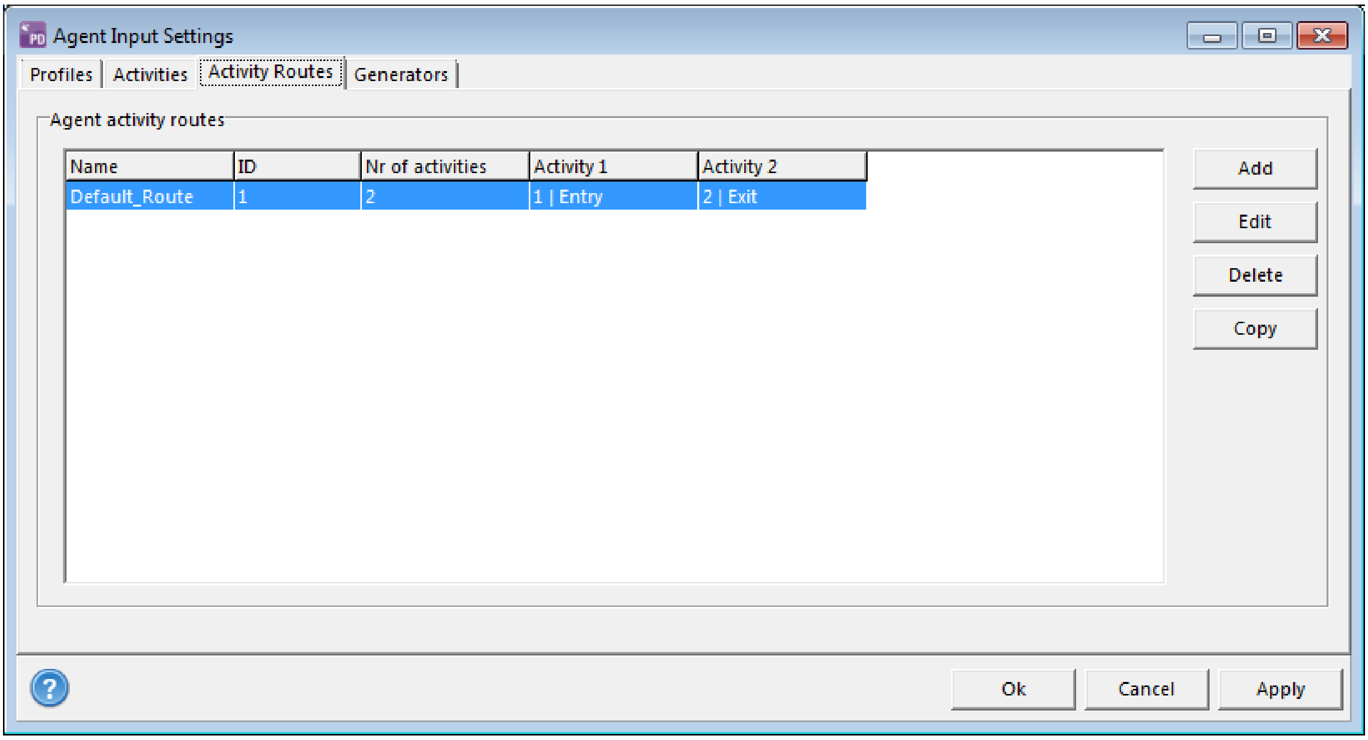


Figure 8: The Activity Routes page of the Agent Input Settings dialog, with one Activity Route.

In the previous two exercises we have created the environment of our pedestrian flow model. The next step is to model the behavior of the pedestrians. To do so you need to define which activities pedestrians will perform and in which order. In this exercise all pedestrians perform the same three activities. The first activity is to enter the tunnel. This typically takes place in an Entry/Exit area. Recall that we have created two Entry/Exit areas through which the tunnel can be entered. The second activity is to visit a shop. We have created two Activities Areas of type Commercial facility where this 'shopping' activity can take place. The third and final activity is to leave the tunnel. The Agents are allowed to leave through the same Entry/Exit Area they used to enter the tunnel.

Exercise 3

Create Agents that enter the tunnel via either the East or West entry, visit one of the shops according to some probability distribution and then leave the tunnel via the other exit.

Above we saw that by default there are only two Activities defined for the agents, an Entry and an Exit Activity. To make the Agents visit one of the shops, we have to create a new Activity which we will call Shopping, and add it to the Activity Route of the agents.

Open the Agent Input Settings dialog and go to the Activities page. Click Add, the Agent Activity dialog will appear. Here we can define our new Activity. Give it the name ‘Shopping’. This Activity can take place in any of the two Commercial facilities that we have created in our environment and not in for example Entry/Exit areas. Therefore select Commercial facility as the Activity type and set the Activity group to ***ALL*** , see Figure 9, then click Apply.

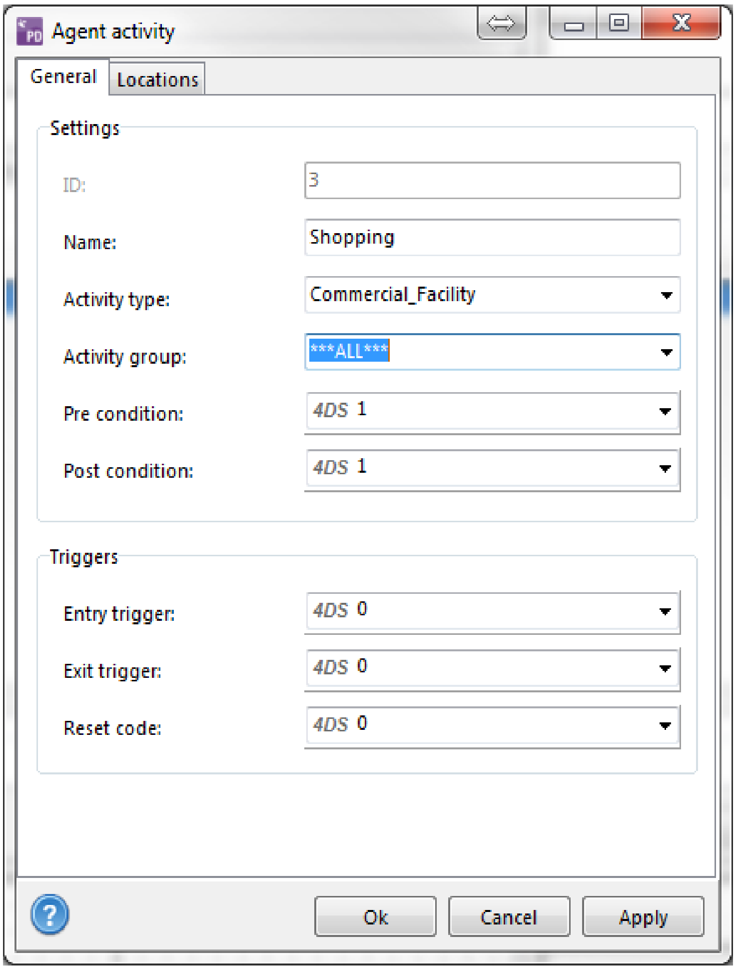


Figure 9: Agent Activity dialog.

We have now created the new activity Shopping. If an agent performs this activity it must be at an Activity Location of type Commercial facility. We have set the Activity group to ***ALL*** which means that the agent can choose to perform this activity in any of the Commercial facilities that we have created in our environment. By default the agent will choose one of the Activity Locations that belong to the Activity group by means of a Uniform distribution. Recall that we created two Commercial facilities in our model, a gift shop and a flower shop. There are thus two locations within our environment where an agent can perform the shopping activity. For this exercise, we will use the default settings. This means that for each Agent that has the Activity shopping in its Activity Route it will have a 50% change to visit the flower shop and also a 50% change to visit the gift shop.

Click on the Ok button of the Agent Activity dialog. Now you will see that a third row has been added to the table of Activities, see Figure 10.

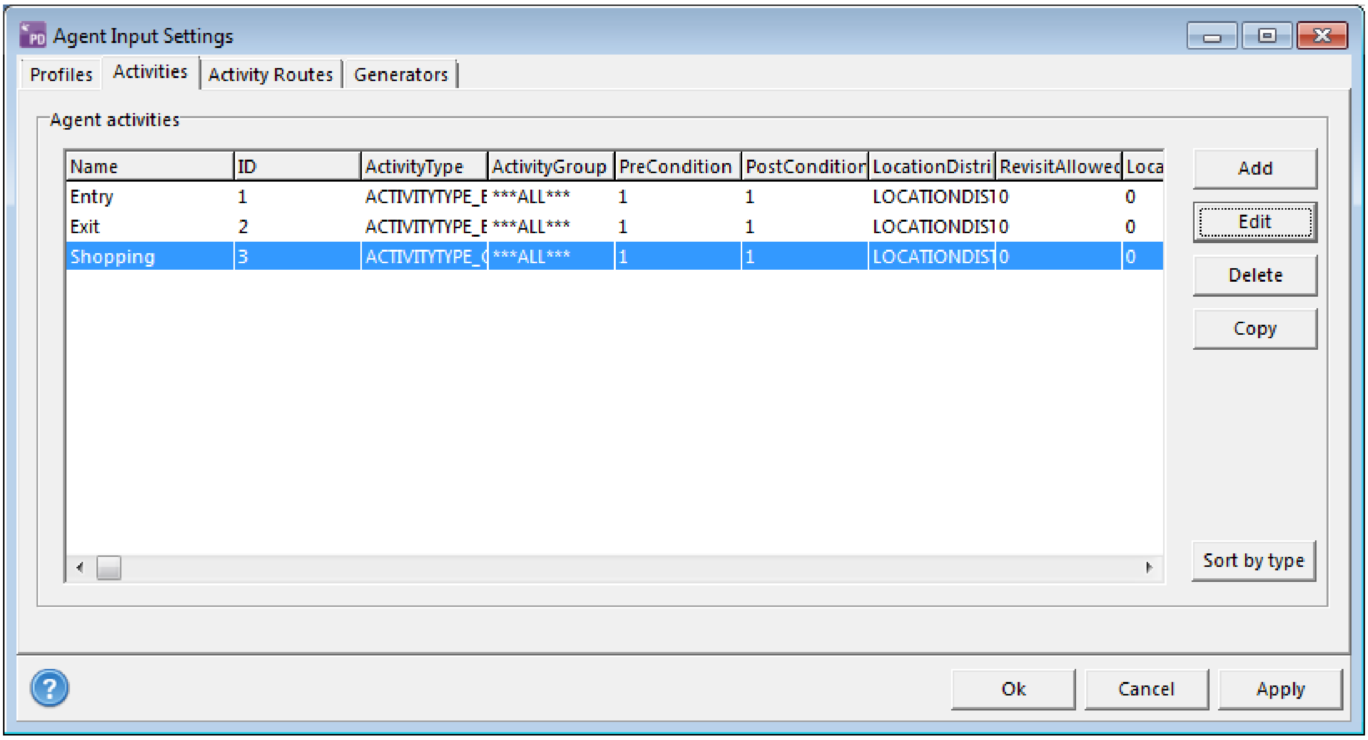


Figure 10: The Agent Input Settings dialog showing the Activity page, three Activities are defined: Entry, Exit and shopping.

An Activity will only be performed by an Agent if it is part of its Activity Route. Switch to the Activity Routes page. Select the Default_Route and click Edit a new dialog will appear in which you can change the route, see Figure 11.

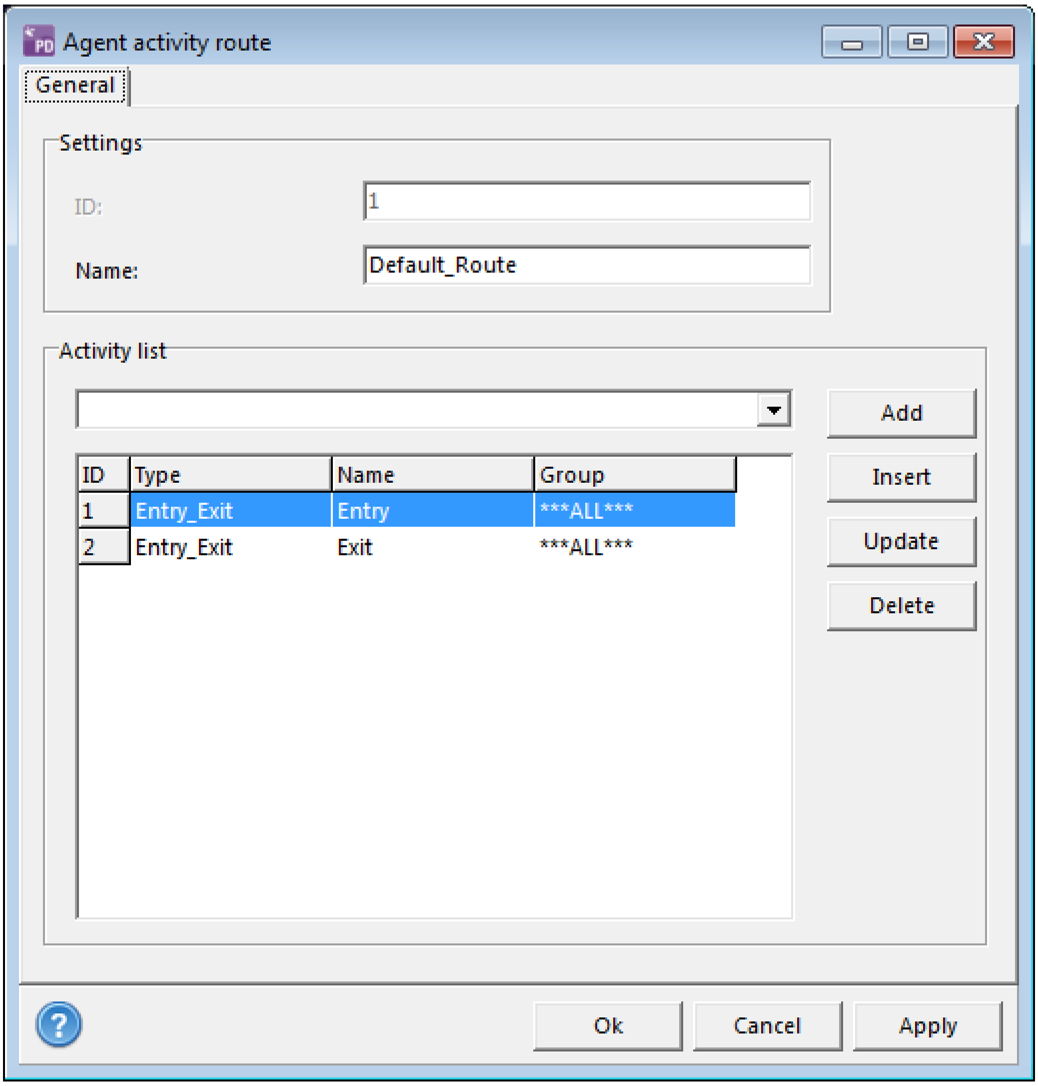


Figure 11: In the Agent Activity route dialog, you can create/alter an Activity Route.

Select the shopping Activity in the selection box and insert it in the list between Entry and Exit, see Figure 12. If you click Add, a new Activity of the currently selected type will be added to the bottom of the list. If you click Insert, the Activity will be inserted directly above the currently selected row of the Activity list.

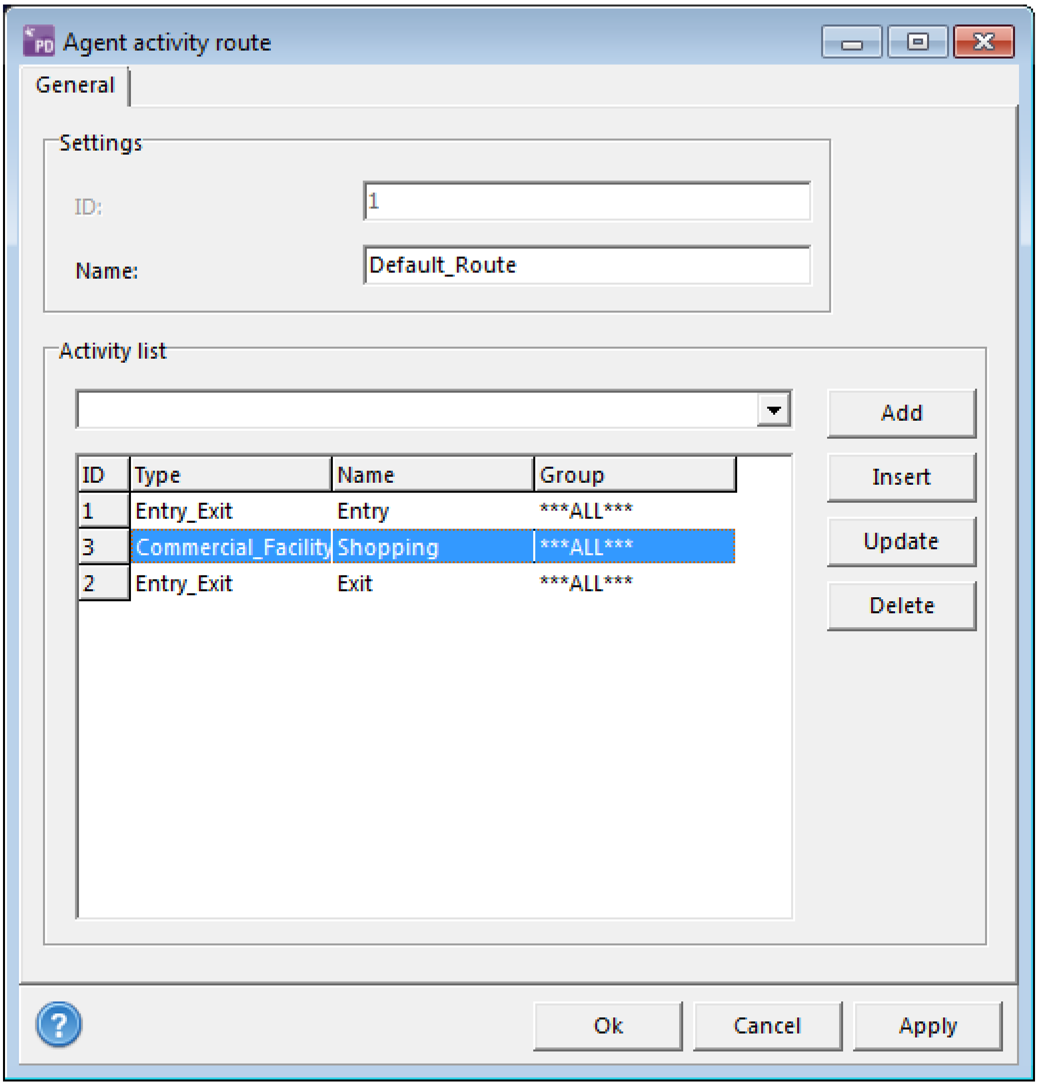


Figure 12: The shopping Activity is inserted in the Activity Route.

Now click Ok in the Agent Activity route dialog and subsequently click Ok in the Agent Input Settings dialog.

When agents plan to visit an Activity Location by default they will choose a random point within the location. Agents will walk to that point. If the agent has reached that point a new location will be determined based on the activity route or the agent will stay in that position for some time depending on properties of the location such as the *activity time*. Since part of the Commercial facilities overlap with the obstacles we need to make sure that all points of the Shops can be visited. Double click on the Commercial facility to open its property dialog. Switch to the Location page and enable *Ensure walkable area*. Do the same for the other shop. If you do not enable *ensure walkable area* some agents will lockdown because they want to visit part of the facility located on the obstacle representing the shop and they thus cannot find a path to reach that point.

Run the model and see what happens.

You will see that an Agent comes from an Entry, then visits a Commercial facility and finally heads for an Exit, as prescribed by the Activity Route!

More precisely, the agents have three activities in their Activity Route. The first Activity is the Entry activity, which can take place in either of the Entry/Exit areas. The second activity is the Shopping activity that we created in this exercise. This activity can take place in any of the Commercial facilities in our model. The third and final activity is the exit activity that can by carried out at either of the Entry/Exit areas. For each of the activities the agents always have two choices. In general if there is more than one Activity Location where an Activity can be carried out, the Agent will pick one dynamically. By default the agents are distributed uniformly over the Activity Locations. In this case it means that the agents have a 50/50 chance to enter the tunnel through either of the Entry/Exit areas. After the agent has been created at one of the Entry/Exit areas its Activity route is checked to determine the next destination. This is the shopping activity which can be carried out at all of the Commercial facilities in our model. Thus the agent also has a 50/50 chance to go to either of the shops.

One would expect that the agent also has a 50/50 chance to pick either of the Entry/Exit areas to leave the tunnel, but this is not the case. Run the model again and watch closely: the agents always use the opposite Entry/Exit area to leave the tunnel. Later we will explain why this happens.

3.5 Generating Agents

We have created the environment of the tunnel and created an Activity Route such that the pedestrians visit one of the shops. The next step is to specify how many pedestrians enter the tunnel. This is specified in an Agent generator. It is used to specify when and what type of agents are created and which route plan they will follow. The generator contains an arrival list indicating when specific agents must be created.

Exercise 4

Every two seconds a pedestrian enters the tunnel to visit one of the shops.

Open the Agent input dialog and switch to the generators page. Select the Default_Generator and click Edit, the Agent generator dialog appears. On the Arrival List page, set the creation time of the Agent to two, see Figure 13. You can edit the values directly in the table. After you have made the changes click Ok and again Ok in the Agent input settings dialog. If you click Edit and use the edit fields in the top of the window than do not forget to press Update before you press OK to close the Agent Generator window.

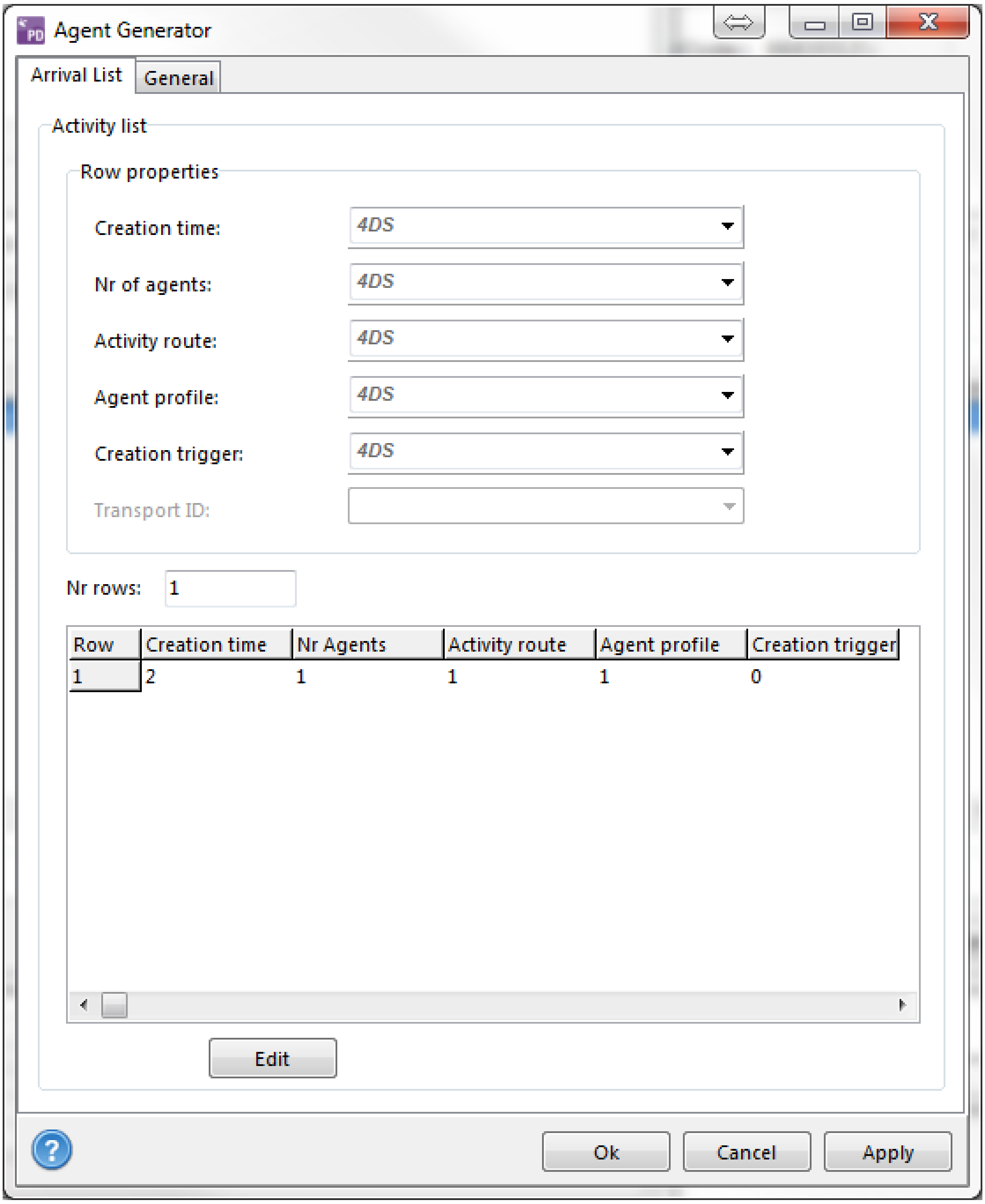


Figure 13: The Arrival list page of the Agent Generator dialog.

So far we have only viewed our model in the 2D model builder, the Model Layout. To see how your model looks in 3D, switch to the Display tab of the main menu. Click on 3D Viewer, a new window appears, see Figure 14. Hold both the left and right mouse buttons and move the mouse up and down to zoom in and out. Run the model again to see the pedestrian walk through the tunnel.

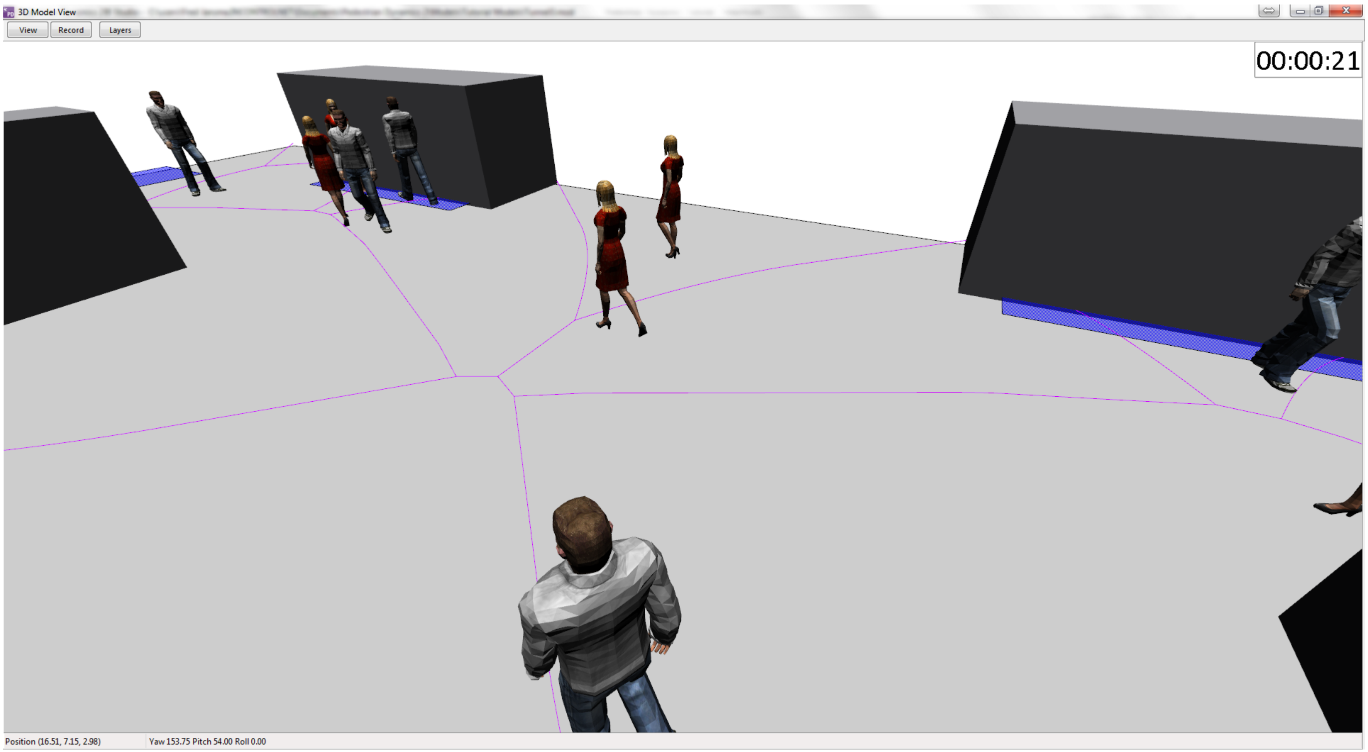


Figure 14: 3D Model view window.

Before we finish our pedestrian flow model of the tunnel we first take a quick tour how to run an experiment and examine the output. Before we can examine the output we first have to run an experiment to collect data.

Exercise 5

Run an experiment and examine the density and frequency map.

Go to the Simulate page of the main menu and click *Wizard* in the Experiment group. A new dialog opens. Click Next twice and then click Start Experiment. Wait until you get the message "Scenarios Completed!". Click Ok. Go to the Results tab in the main menu and open the 2D output window, see Figure 15. On the left the Draw toolbar is visible and on the top you can see the Statistics toolbar. The last four icons in the Statistics toolbar are used to draw the density map, draw the frequency map, travel times map and to clear the layer, respectively.

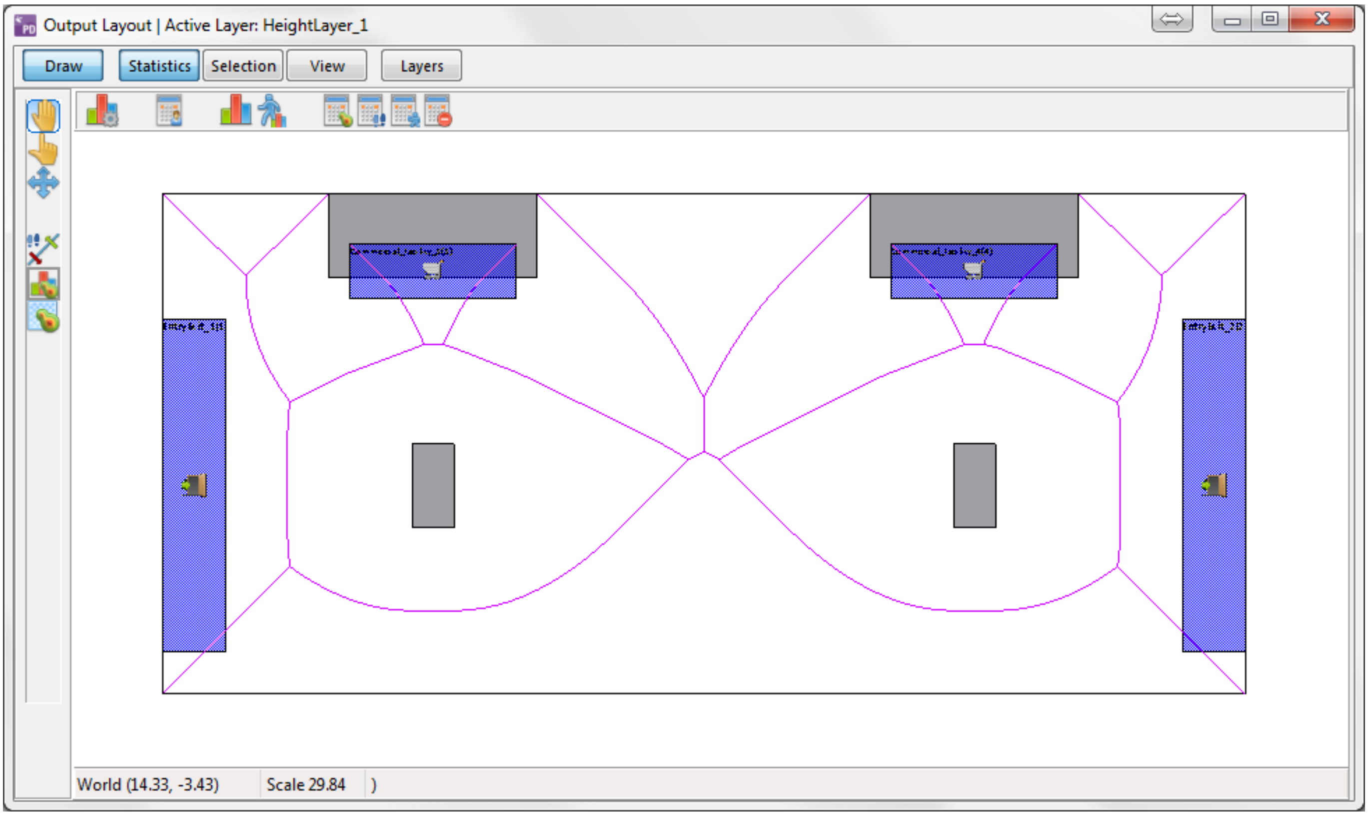


Figure 15: The 2D output window

Click on the Density map Icon, a new dialog appears. Click Calculate, the density map will now become visible in the Output window. The result should be similar to Figure 16. The Density map shows the maximum density that occurred during the simulation. Note that most of the area where the agents can walk is colored light blue, this means that the density was never very high. On the North side of the tunnel between the shops and the columns the area is partly light green. In this area the density has been higher but was never very dense.

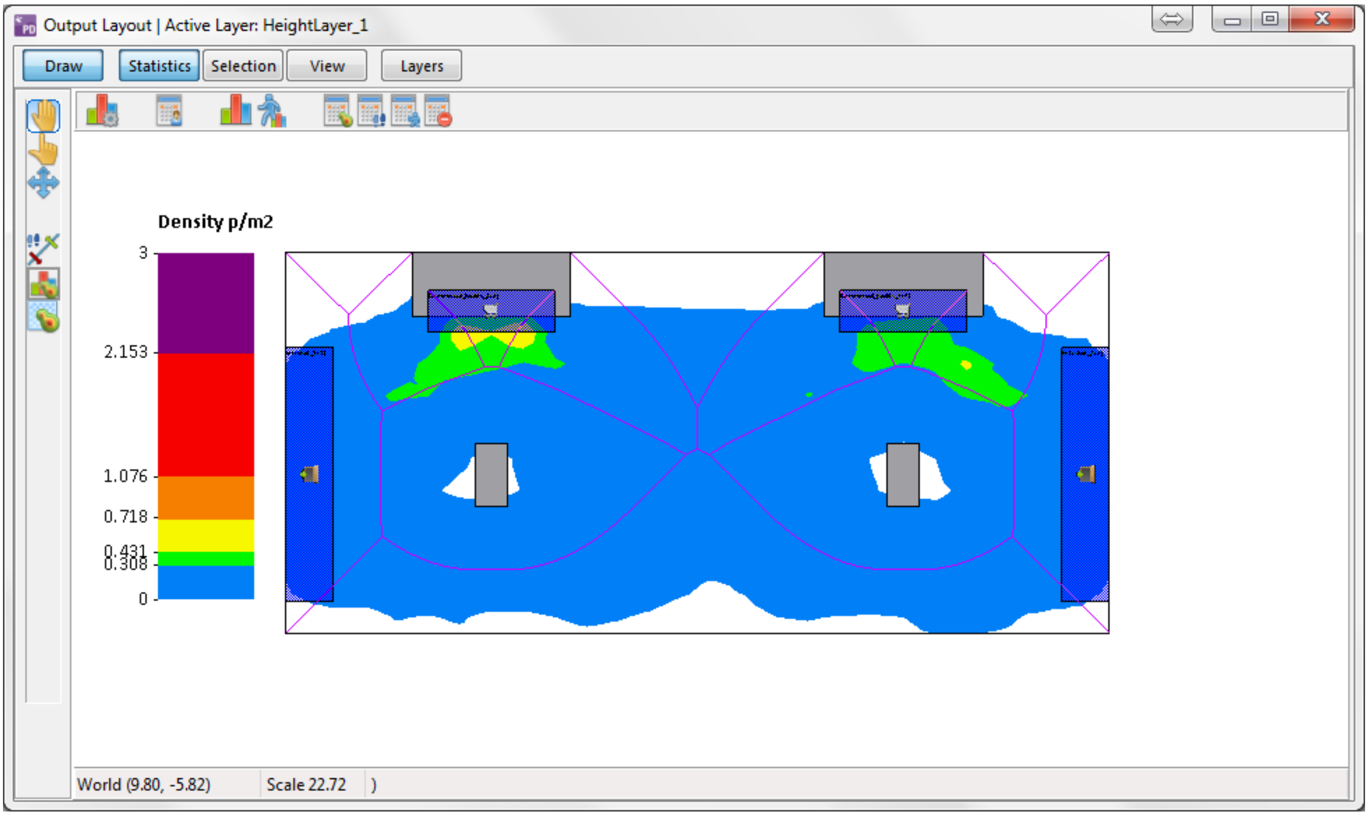


Figure 16: The 2D output window showing the density map.

Click the last Icon in the Statistics toolbar to clear the density output. Do the same to create a frequency map. The frequency map is simply a count of the total number of pedestrians that walked through the area during the simulation.

3.6 Agent profiles

When you create a pedestrian flow model of a real system you will typically distinguish several groups of pedestrians that use the environment in different ways. These groups might have different activities they want to perform in the environment. For example, in our tunnel case not all of the pedestrians will use the tunnel to visit one of the shops. We have seen above that this can be modeled by creating different Activity routes. The Generator can create agents and assign them to different Activity Routes.

There are also other differences between Agents, such as their walking speed and their 2D and 3D visualization. These differences belong to the Agent Profile. One common application is to create separate profiles for men and women with different looks. These can then be used in combination with different Activity routes which visit for example the Man and Women's locker rooms, respectively.

In our Tunnel case, we will use Agent profiles to visually distinguish pedestrians who use the tunnel only to cross the railway from pedestrians that visit the shops.

Exercise 6

Not all of the pedestrians that use the tunnel will visit one of the shops. Create two agent types Shoppers and Passers. The Passers enter the tunnel on either side and leave on the other side. The Shoppers enter the tunnel and visit one of the shops. 80% of the Shoppers leave the tunnel using the west exit. This means that some of the Shoppers leave the tunnel on the same side they used to enter.

We have to create a new Agent profile but we will first change the name of the existing one. Open the Agent input settings dialog. Select, in the Profiles tab, the first row in the Agent profiles table and click Edit. The Agent profile dialog opens, see Figure 17. Change the name Default_Profile into Passers and click Ok.

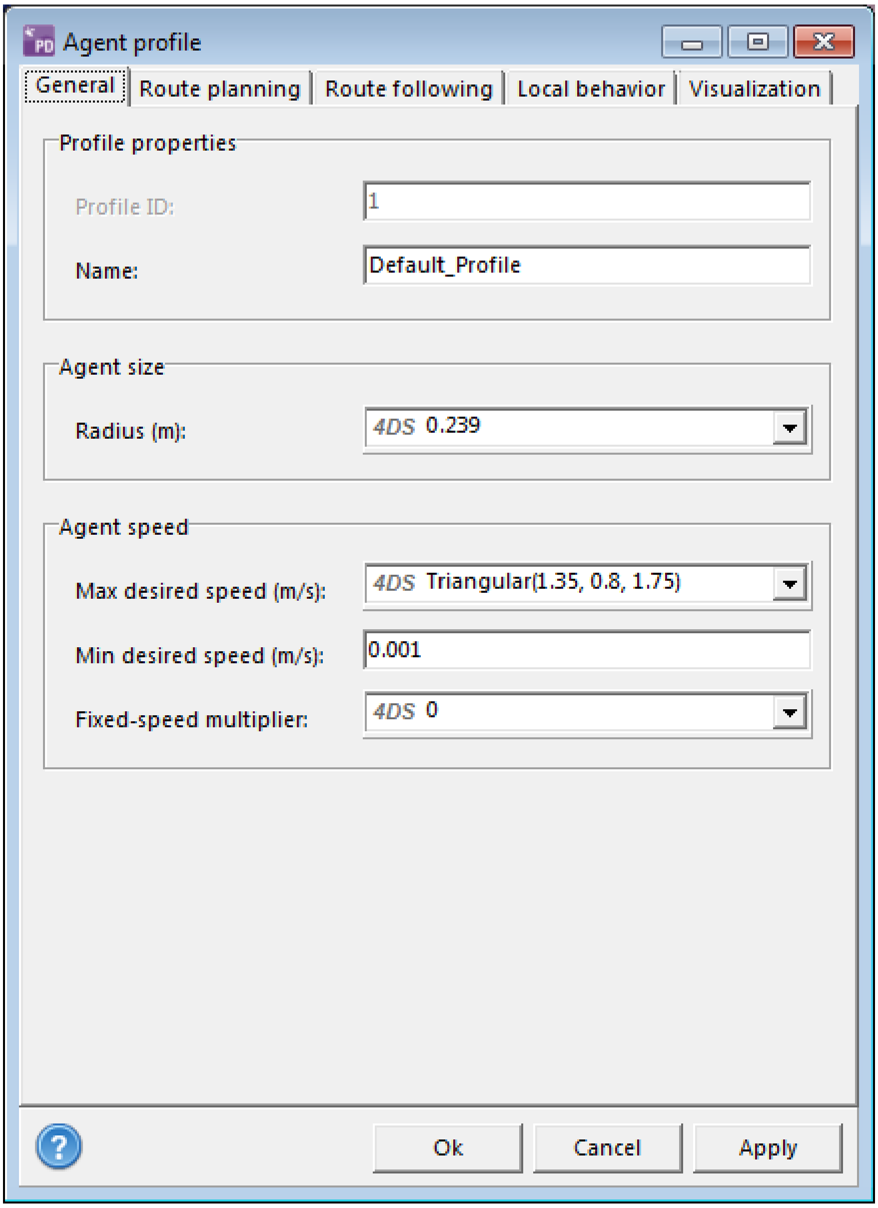


Figure 17: The Agent profile dialog.

Click Add to add a new Agent profile. In the Agent profile dialog type Shoppers as the name for this profile. Furthermore, switch to the Visualization page and change the color to Red, see Figure 18. This way, we can distinguish between the Agent profiles during a simulation run, Passers blue vs. Shoppers red.

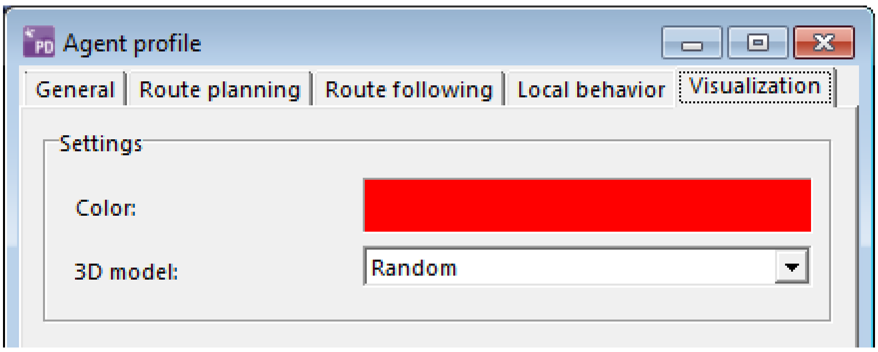


Figure 18: The Visualization tab of the Agent profile dialog.

From the exercise description, we learn that Passers enter the tunnel from one side and leave via the other. Shoppers also enter the tunnel via one side, then visit a shop, but subsequently sometimes leave the tunnel via the same side they used to enter. How do we put this information into the model?

As we have two Agent profiles with each a different usage of the tunnel, it makes sense to use a different Activity Route for each Agent profile. For Shoppers, the Activity Route must contain three Activities: Entry, shopping and Exit. For Passers, the Activity Route must contain two Activities: Entry and Exit.

Do we have any restrictions for the locations of the Activities? Yes. For Passers, the Exit location must not be the same as the Entry location. For Shoppers, the Exit location may be the same as the Entry location for some of the Shoppers.

During exercise 2 and 3, we already saw that with the default settings, the Agents always used a different Entry and Exit location. We will now take a closer look at the location settings of the Exit Activity. You can do this in the following way: open the Agent Input Settings dialog, select the Activities tab, select the Exit Activity, click Edit and select the Locations tab, see Figure 19.

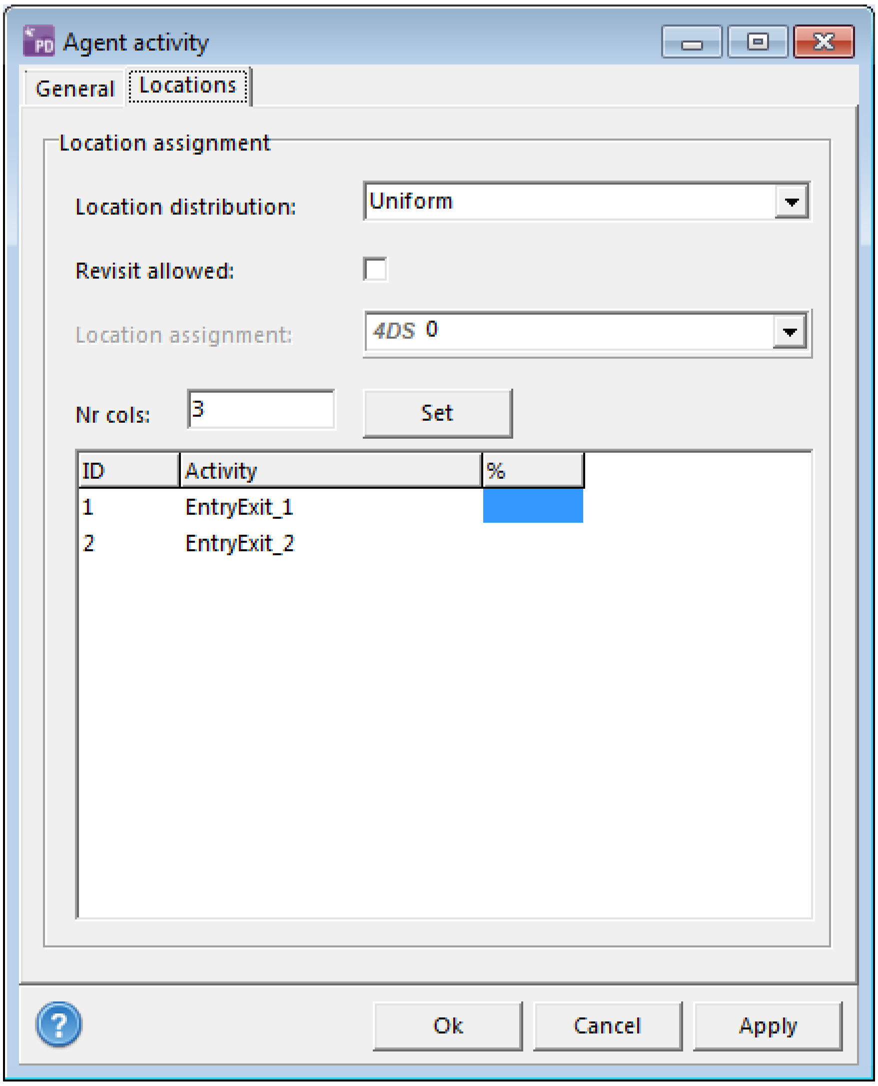


Figure 19: The Locations tab of the Agent Activity dialog, Revisit allowed is unchecked.

Notice that the Location distribution is set to Uniform. You thus would expect that the agents have a 50/50 chance to end up in either Entry/Exit area. This does not happen because the "Revisit allowed" checkbox is unchecked. If Revisit allowed is unchecked for a particular activity it means that the agent is not allowed to perform that activity at an Activity location that the agent already visited. In our example this means that if the Entry Activity was performed at EntryExit_1, the Uniform distribution will be overruled when it selects the same Activity Location for the Exit Activity, because revisit is not allowed! Therefore, the Agent will always have a different Entry/Exit area to perform the Entry and Exit Activity.

In our current exercise, the situation as described above is the same for the Passers. Recall that 80% of the Shoppers will leave the tunnel from the side they used to enter the tunnel. Therefore we need to create a new Activity. We will name this activity: Exit_Shoppers.

Open the Agent Activity dialog and create a new activity with the name Exit_Shoppers. Select as type Entry_Exit and as group ***ALL***. Switch to the Locations page and check Revisit allowed, see Figure 19. The agents will now have a 50/50 chance to go to either of the Entry/Exit areas in our model.

> Besides the uniform distribution other options for the Location distribution are an Empirical distribution or a UserDefined distribution. With the Empirical distribution, you can enter a percentage in the third column of the location table. With the UserDefined distribution you can select a predefined logic, such as picking the closest or fastest option, or distributed based on surface area. Advanced users can alter these logics or enter their own code in the 4DScript editor.

If we want to make sure that 80% of the shoppers choose the West exit we have to change the Location distribution. Change the Location distribution to Empirical, a table appears on the location tab. You can now change the percentages in the table. Set the West Entry/Exit area to 80% and the east to 20%, see Figure 20.

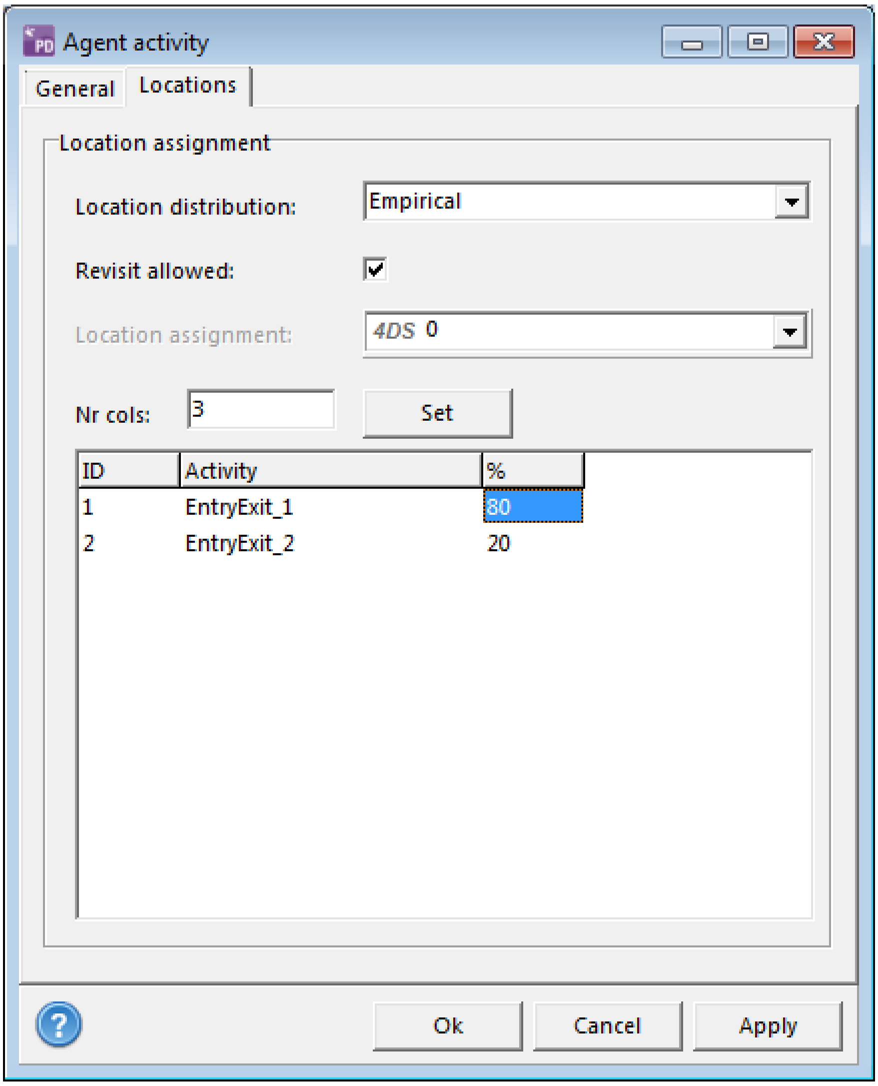


Figure 20: The Locations tab of the Agent Activity dialog.

Now we have defined all the Activities necessary to build the Activity Routes for the Shoppers and Passers. Go to the Activity Routes page and create two Activity Routes: Passing_Route and Shopping_Route. Use the correct Activities for each Route. The result should be the same as Figure 21.

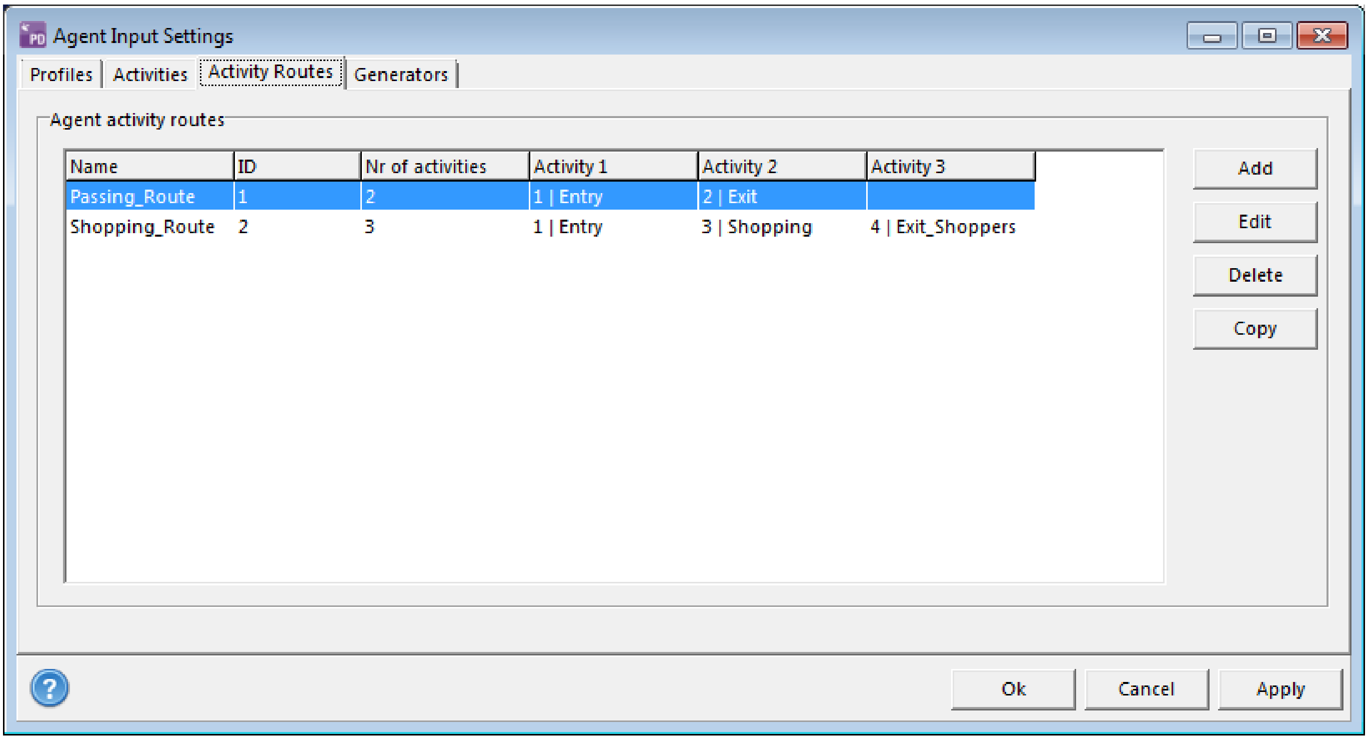


Figure 21: The Activity Routes page of the Agent Input Settings dialog.

Now we only need to adjust the Agent Generator, which is responsible for creating the Agents, so that both Agent profiles are created. Go to the Generators page in the the Agent Input Settings dialog. Select the first row in the Agent generators table and click Edit. Click Add to add a new row in the Arrival List. Set the row properties in order to let the Generator create the two Agent Profiles each with its Activity Route. The results should be similar to Figure 22. Click Ok and again Ok in the Agent Input Settings dialog. Now run the simulation and see what happens.

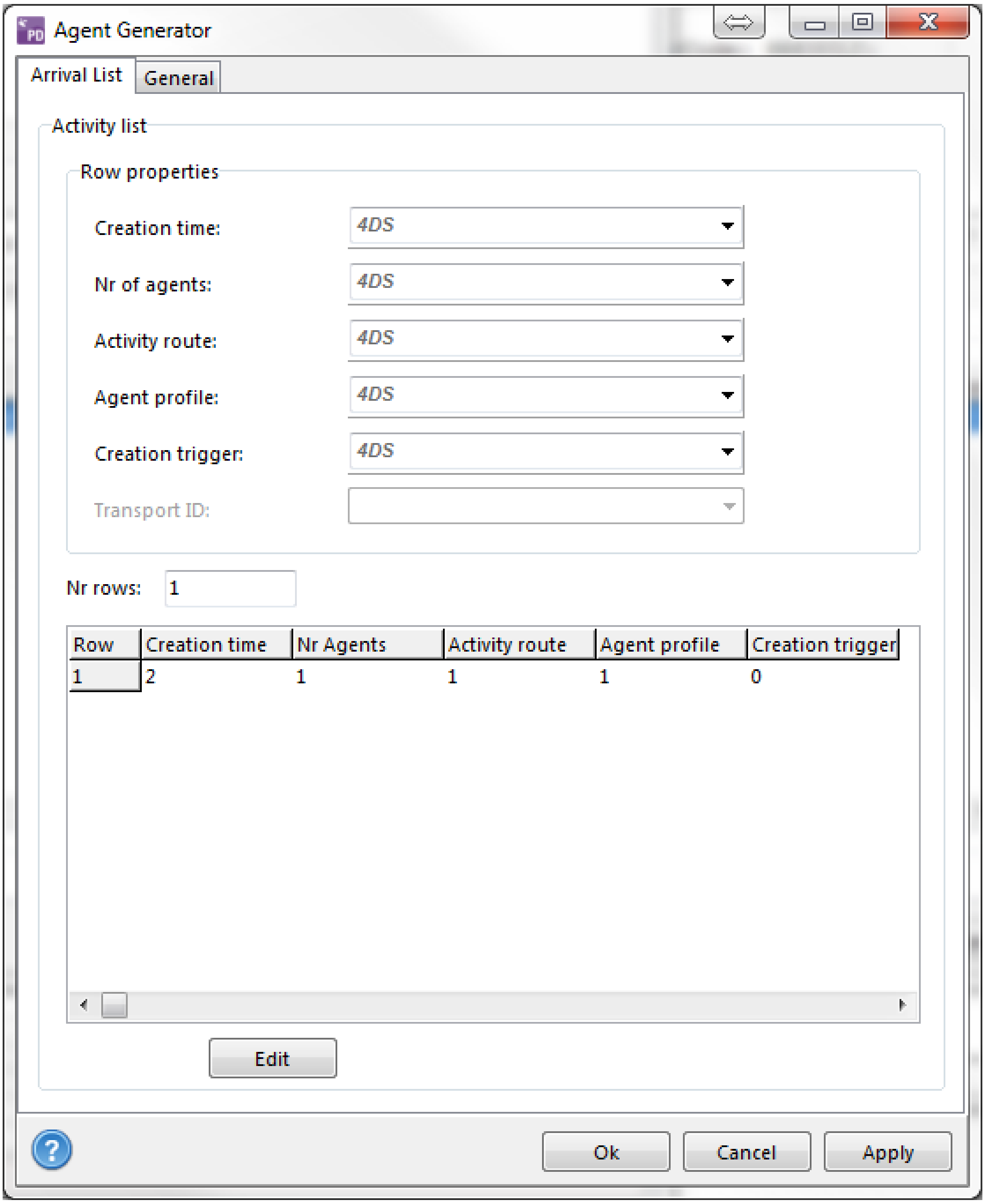


Figure 22: The Arrival List page of the Agent Generator dialog.

With these settings, we see that most of the Shoppers leave the tunnel via the West side of the tunnel.

3.7 Working with Layers

Recall that this tunnel not only connects the center of town and the West part of town, but also functions as a train station with railway tracks above the tunnel. In order to integrate this in our model, we need to add a second Height layer that represents the railway track level. We also need to create a connection between the tunnel level and the railway track level.

Exercise 7

Create a second Height layer that represents the railway track level. Create a flight of stairs and an escalator that will act as the connection between the tunnel level and the railway track level. Also create arriving and departing passengers.

Draw a rectangular elongated Height layer. Give it roughly the same dimensions as HeightLayer_1, but position it perpendicular to HeightLayer_1, see Figure 23. The track level is located 4 meter above the tunnel level therefore we need to set the z location of the layer to 4. Switch to the Select mode and double click the new layer, the properties dialog of the layer will appear. On the general page set the z-loc to 4.

To easily manipulate a particular layer it can be useful to make the other layers invisible. Click Height layers in the menu of the Model Layout window. On the right side of the window, a Height layer settings panel appears. The 'eye' icon before the name of a Height layer allows you to toggle the visibility of that layer on and off. You can only manipulate one layer at a time. This is called the Active layer. If you for example draw a new Obstacle it will be added to the active layer. The active layer's bar is marked with light blue. If you click another layer's bar it will become the active layer, see Figure 23.

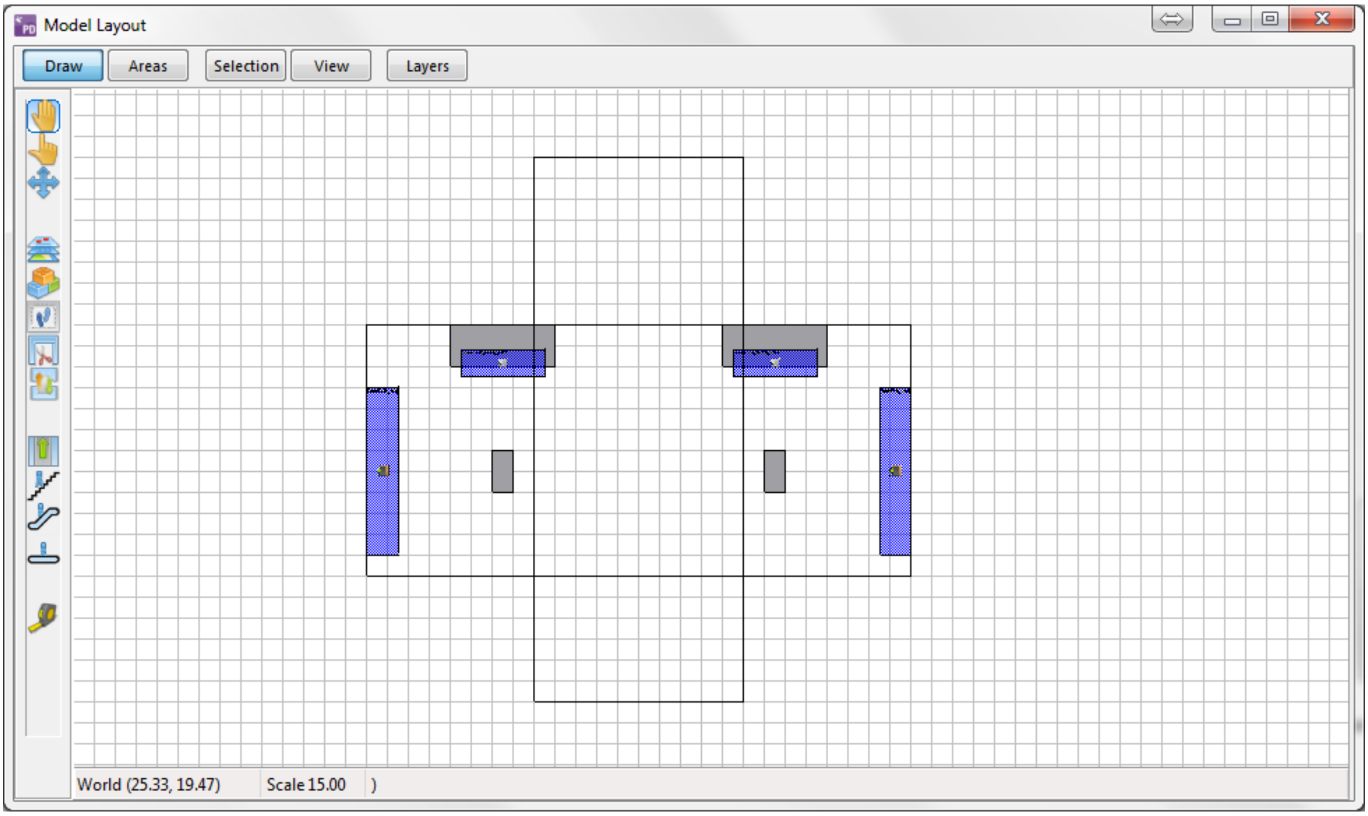


Figure 23: The Model Layout window with the two Height layers. The Height layer panel is visible and HeightLayer1 is the active layer.

The next step is to build a connection between the two Height layers, such that agents can walk from one layer to another. There are several infrastructural elements in PD that can form a connection between Height layers that have a different z-location. In this case the connection is formed by a flight of Stairs and an Escalator, but you can also use a Moving walk.

First, we will build a flight of stairs. In the drawing toolbar select the Stairs icon. Draw a rectangle in HeightLayer_1, make sure that this is the active layer. Notice that the orientation of the stairs is east-west. Note that the East border of the Stairs is colored red. This indicates the bottom of the Stairs and is also the tilting edge of the Stairs element. In this case the Stairs should by orientated south-north and the north side should be the top of the stairs.

To change the orientation of the Stairs select it and press F7. Fill in a rotation value between 0 and 360 degrees. In this case type 270 degrees. Switch to the Move mode to select and position the stairs. You can also press F8 after selecting the stairs and fill in a new location. The stairs do not yet form a connection between the two layers. Therefore switch to the Select mode and double-click the flight of stairs; a properties dialog of the stairs will appear. On the General page check the Transfer box in the connection settings group and set the 'layer from' and 'layer to' properties. Set these to connect HeightLayer_1 and HeightLayer_2. Since the stairs forms a connection between to Height layers we do not have to set a Tilt height for the Stairs. Open the 3D Viewer from the display tab of the main menu to see that the Stairs connects the layer properly.

Now, the same steps can be repeated for the addition of an escalator. Select the Escalator icon in the Draw toolbar and add an escalator. Make sure the escalator forms a connection between the two Height layers. Make sure the escalator is set such that it caries passengers from the tunnel to the station platform.

On the second Height layer we will build two more Entry/Exit areas to simulate an arriving or departing train.

Switch to the Activities toolbar and add the two Entry/Exit areas, the result should look like Figure 24.

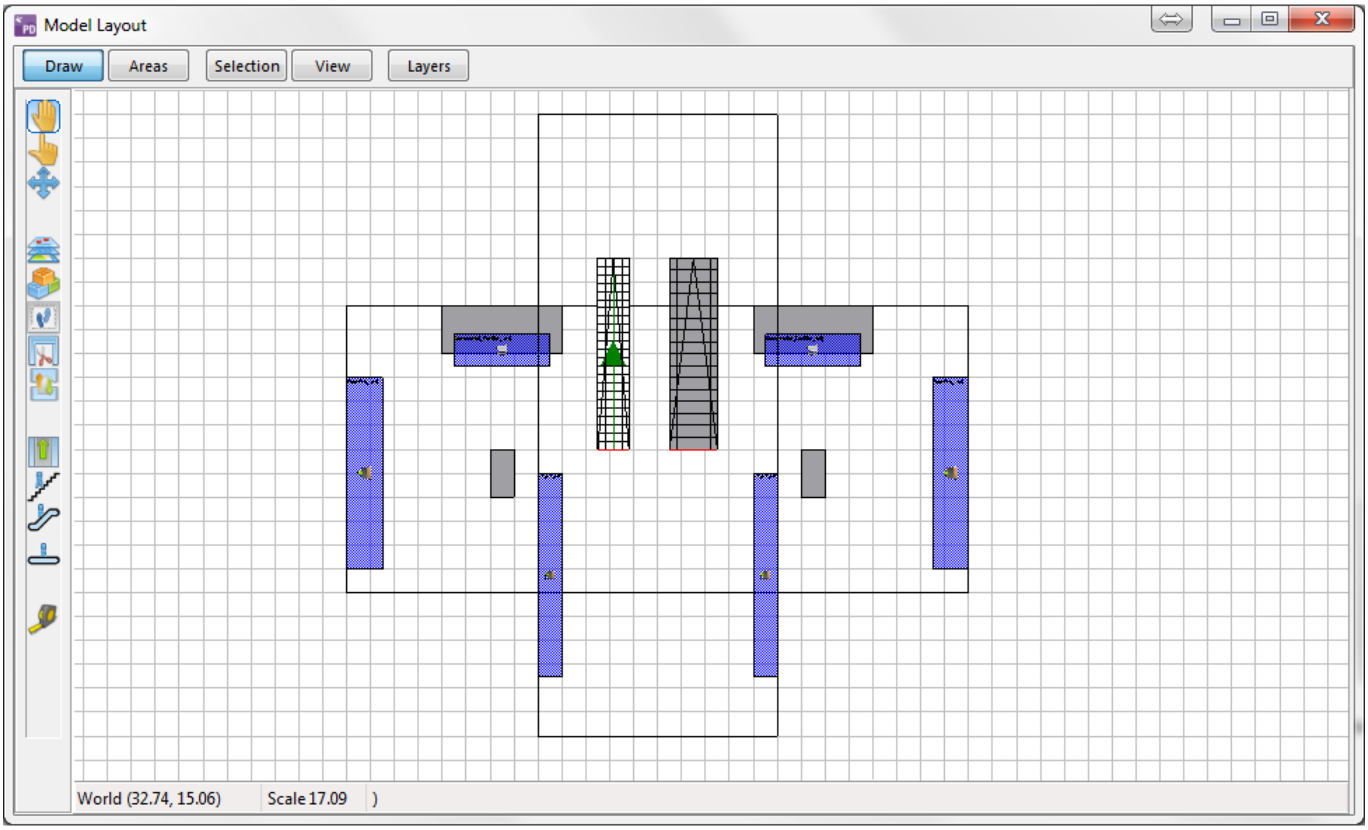


Figure 24: The Model Layout window with the two Height layers an escalator and a flight of stairs form a connection between the two layers.

We also need to model arriving and departing passengers. To be able to distinguish passengers from shoppers and passers we create a new profile Passengers and give them a different color. Open the Agent Input dialog, create a new agent profile passengers and set the color on the visualization tab to green.

The next step is to think about the Activity Routes and Activities of the agents. The passers still go from one Entry/Exit area of the tunnel to the other, but they cannot go to all Entry/Exit areas. They will never visit the Entry/Exit areas on the platform. The shoppers enter through a tunnel Entry/Exit area, visit a shop and exit through a tunnel Entry/Exit area. The passengers will either enter through a tunnel Entry/Exit area and exit from a platform Entry/Exit area or enter from a platform Entry/Exit area and exit through a tunnel Entry/Exit. Until now we created activities that could be performed at all Activity Location of a certain type. For example, the Enter Activity could take place at ***ALL*** Entry/Exit areas that we created in our model. Because we added the Entry/Exit areas on the platform this isn't the case anymore. To model this behavior we can create two groups of Entry/Exit areas, the platform and the tunnel group.

To add an Activity Location to a group, switch to the Select-mode and double click the location. The properties dialog of the location appears. On the General page, the name of a group can be typed. Open the properties dialog of the Entry/Exit areas of the tunnel and give the group the name Tunnel, see Figure 25. Do the same for the Entry/Exit areas of the platform set the name of the group to Platform.

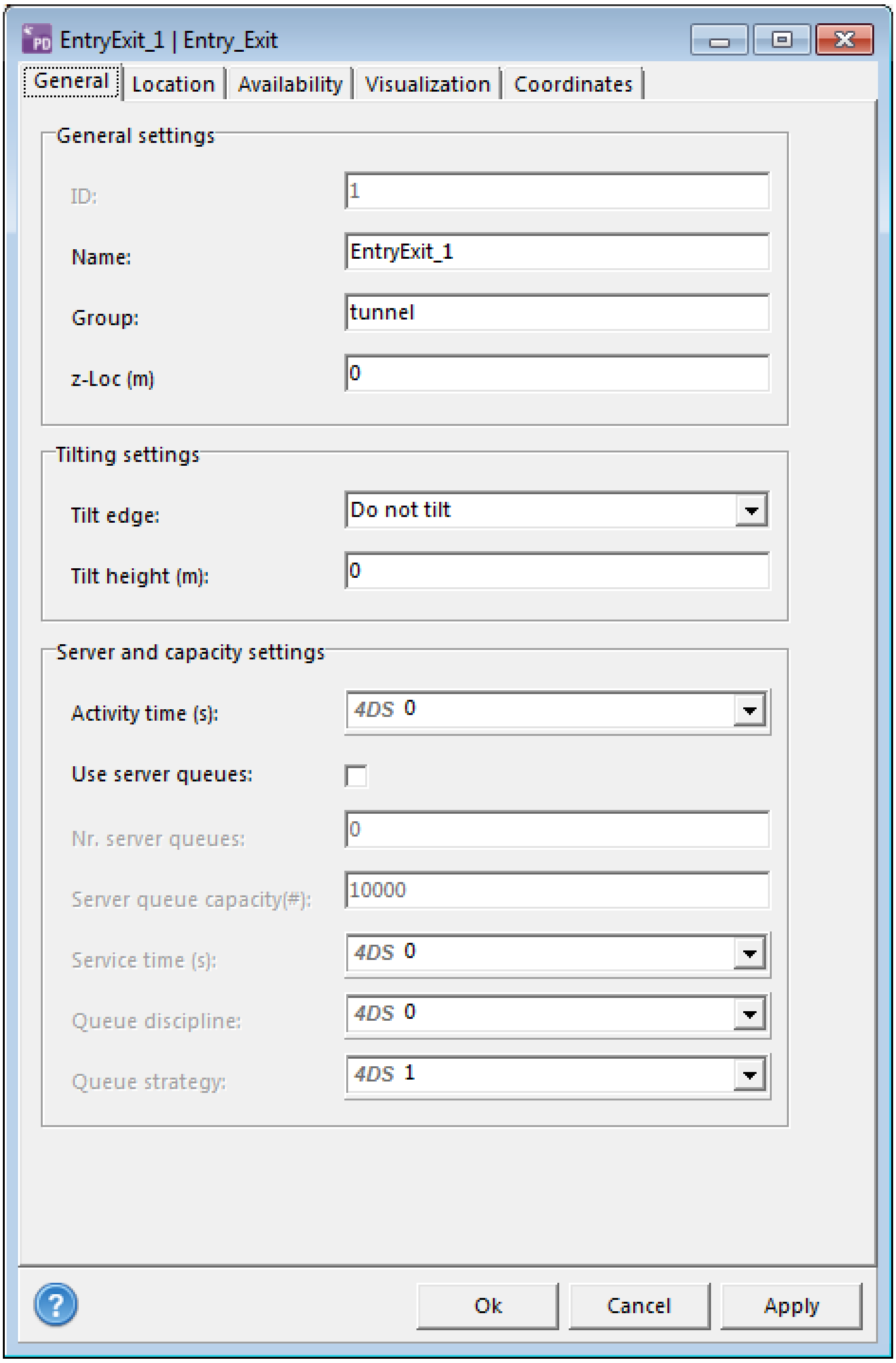


Figure 25: The properties dialog of the Entry/Exit area located on the West side of the tunnel.

Now change the Enter, Exit and Exit_Shoppers activities to make sure that these only take place at the Entry/Exit areas that belong to the group *tunnel*. Open the Agent Input dialog and switch to the Activities tab. Select the Entry activity and click Edit, the Agent Activity dialog appears. Note that in the dropdown list of the Activity group Tunnel and Platform have been added. Change the setting of the Activity Group to *tunnel*, see Figure 26. Do the same for the Exit and Exit_shoppers activity.

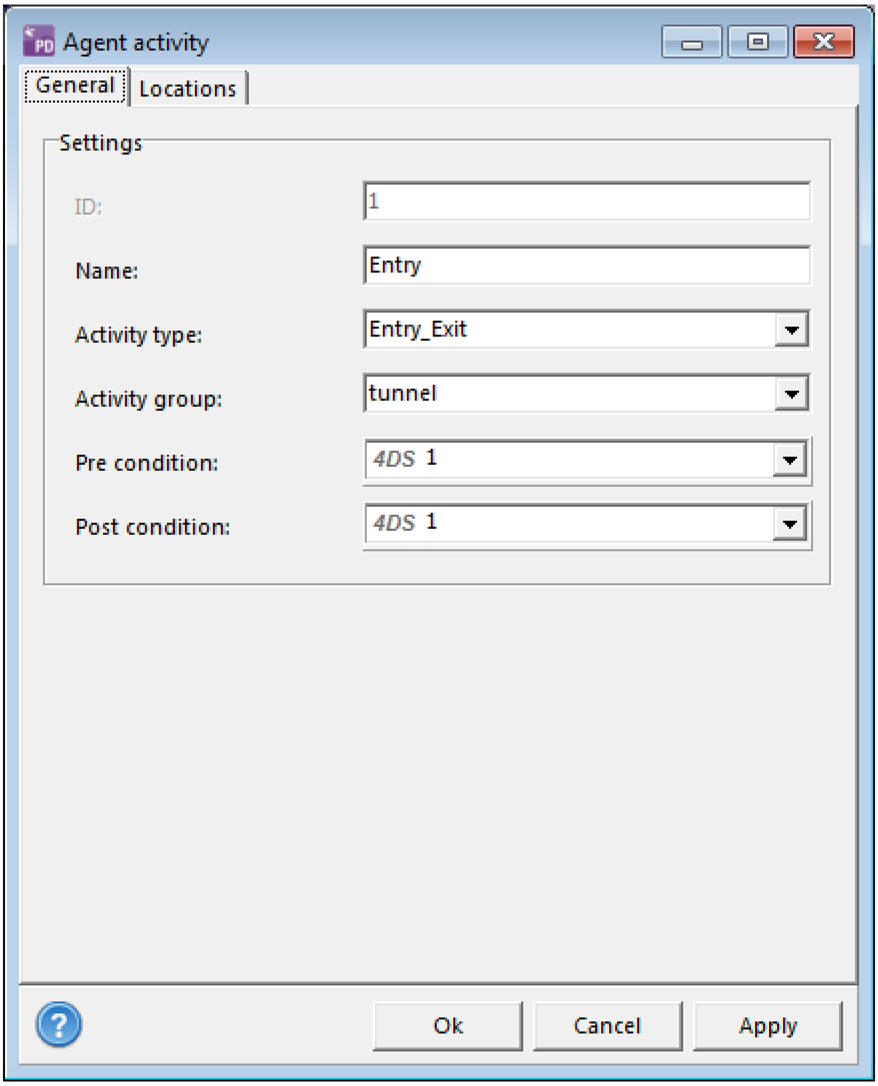


Figure 26: The Agent Activity dialog of the Entry Activity.

To model the Arriving and Departing passengers we only have to add one new Activity. This activity is used to model the arrival or departure of a passenger on the platform. Click Add on the Activities tab of the Agent Input dialog. Give this new activity the name Arrive/Depart. Set the Activity type to Entry/Exit and the group to Platform and click Ok in the Agent Activity window.

There are two Activity Routes that we need to add, one for departing passengers and one for arriving passengers. Add the Depart and Arrive Routes to the Activity Routes, the result should be similar to Figure 27.

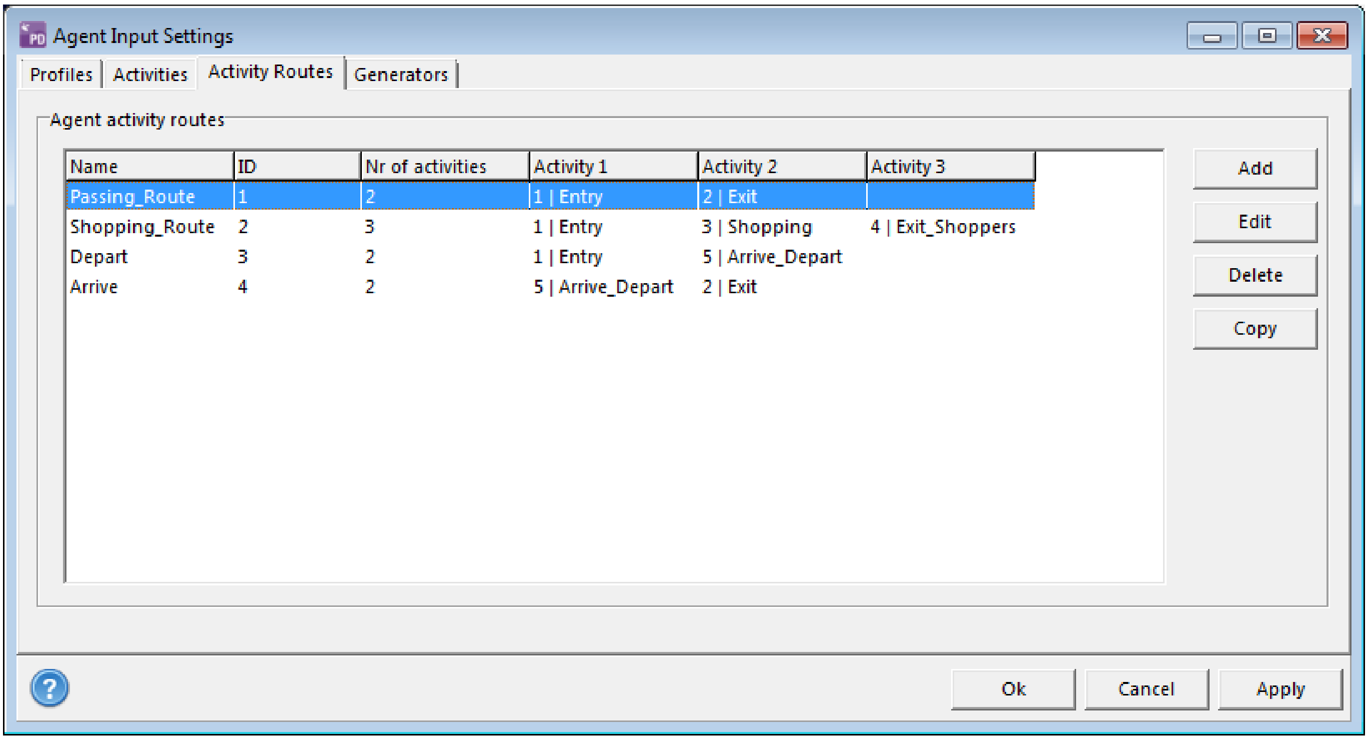


Figure 27: The Activity routes page of the Agent Input settings dialog for the tunnel.

The last step we have to do is to make sure that the generator creates Agents of the type Passenger. We need to create passengers some of which should take the Depart route but others should follow the Arrive route. Change the Arrival list of the Generator and make sure the result is similar to Figure 28.

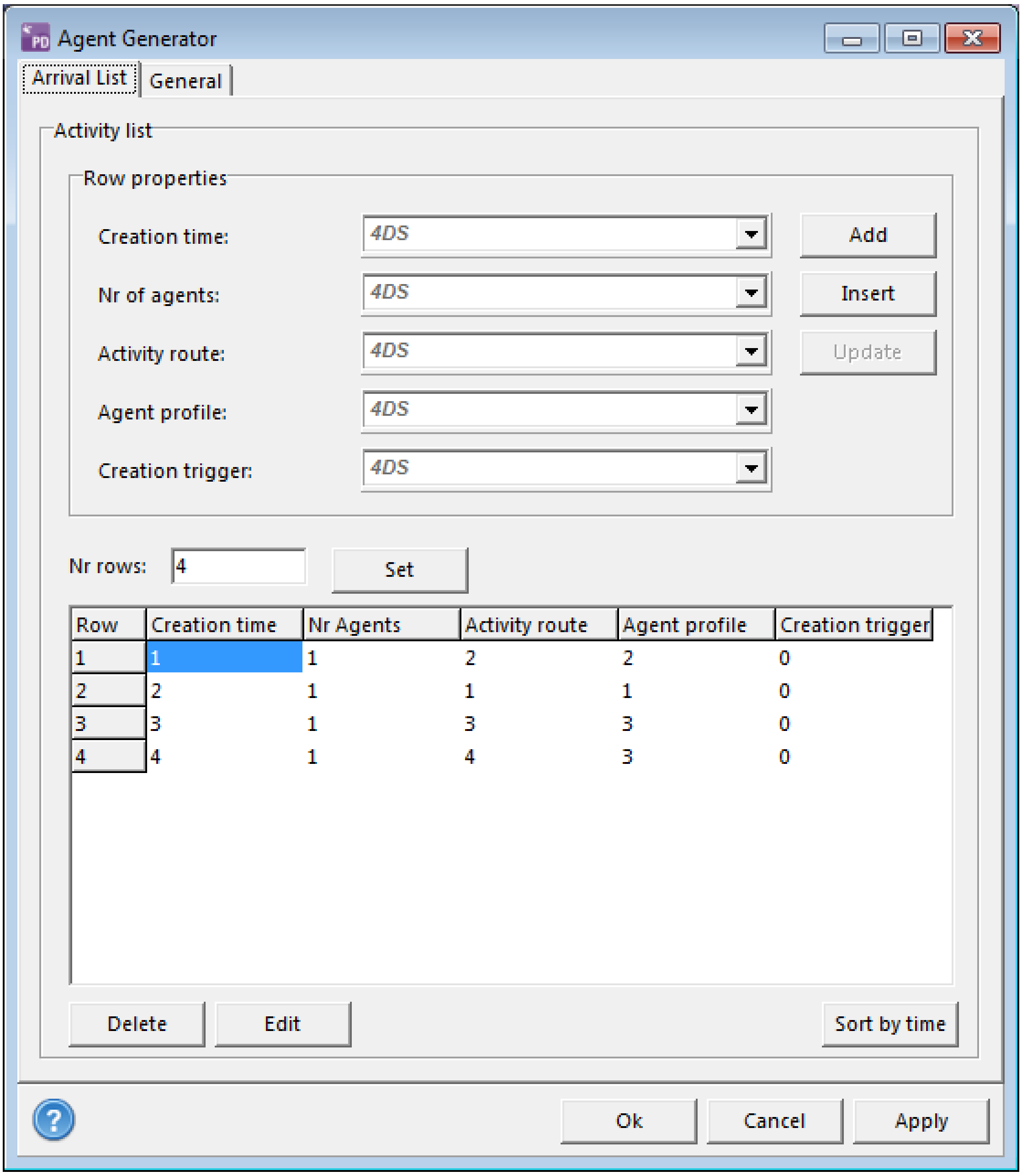


Figure 28: The Arrival list for the tunnel.

The Arrival list displayed in Figure 28 will after one second after the start of the simulation create a single agent of profile type two (Shoppers) that follows Activity route two (Shopping), after two seconds a single Agent of type Passers is created that will follow route 1 the Passing route, at second three a single agent of type 3 (Passengers) is created that will follow route 3 (Depart) and finally at second four a single agent of the type Passenger is created that will follow route 4 (Arrive). This list will be repeated continuous during a run creating more and more agents.

From the Display tab on the main menu open the 3D Viewer window. Run the model and see how the pedestrians walk from the platform level to the tunnel, see Figure 29.

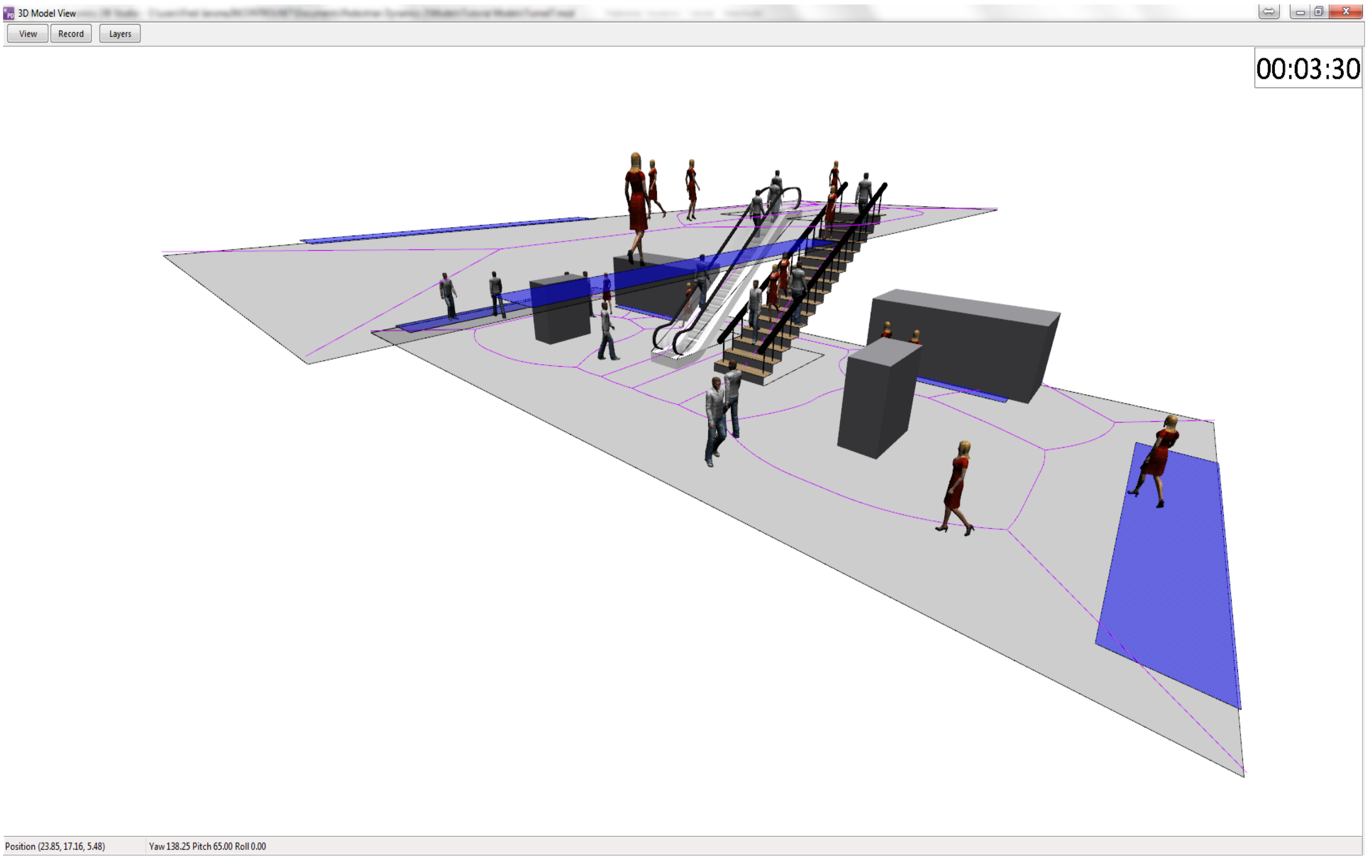


Figure 29: The 3D view window.

Run another experiment and examine the density map to see if the flight of stairs and the escalator have enough capacity for all pedestrians. Open the Experiment Wizard from the Simulate tab of the main menu. Click Next until you can Click 'Start Experiment'. After the simulation has finished switch to the Display tab and open the 2D output window and create a Density map. Click Height layer to open the Height layer options panel, the result should be similar to Figure 30. Especially on the escalator and flight of stairs the density is relatively high.

Note that the densities on the Stairs and Escalator are partly drawn on the map of HeightLayer_1 and partly on HeightLayer_2. The transition area between the two layers seems to have a low density, but this is not the case. It is caused by the fact that data from one layer is not shared with the other. If you would create a separate Height Layer of the Stairs or Escalator and run another experiment then you will see that the density is higher over the whole Stairs and Escalator.

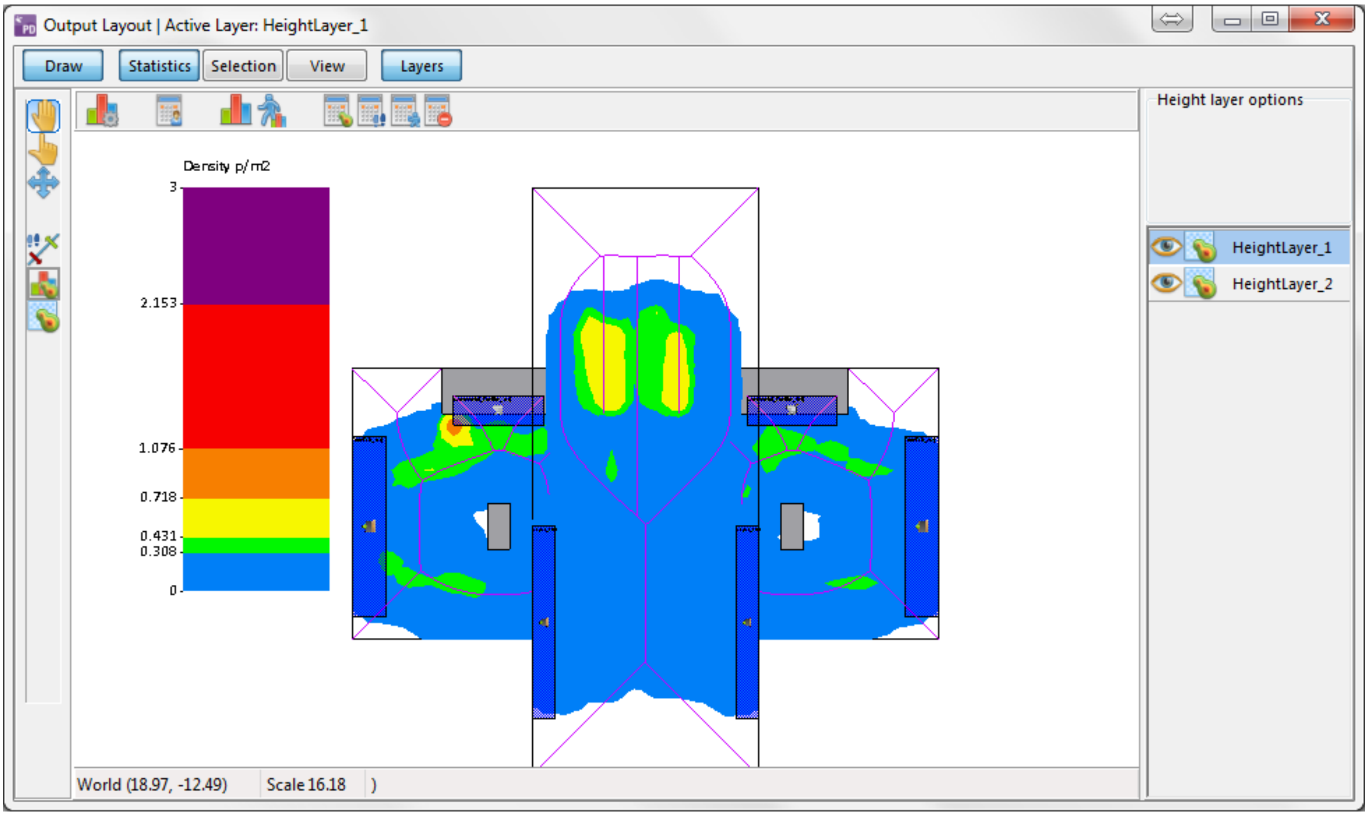


Figure 30: The 2D Output window showing the density map of HeightLayer_2 of the tunnel7 model.

4 Creating an evacuation model

In this section we will build a model of a department store to examine the evacuation time of the building.

The objective is to get familiar with the Action timer element and how elements such as a Passageway and an Escalator can be used in evacuation models and learn more about settings of the Activity Locations. Main goal is to learn how to build an evacuation model in Pedestrian Dynamics.

Case 1: Evacuation of a department store

A manager of a department store wants to investigate the evacuation time of the store and train his staff for an emergency evacuation. The department store has four floors with several departments selling different kinds of products, see Figure 1.

Each day there are large numbers of people in the department store that are not very familiar with the layout of building. This makes it difficult and expensive to set up a proper fire drill. The manager wants to train the staff showing them different simulation scenarios with different emergency conditions so that they are better prepared to assist evacuating the customers during an emergency evacuation. Questions he has are; What is the evacuation time if everything goes as planned? What happens if a fire prevents access to one of the escalators? Can the building still be evacuated in reasonable time?

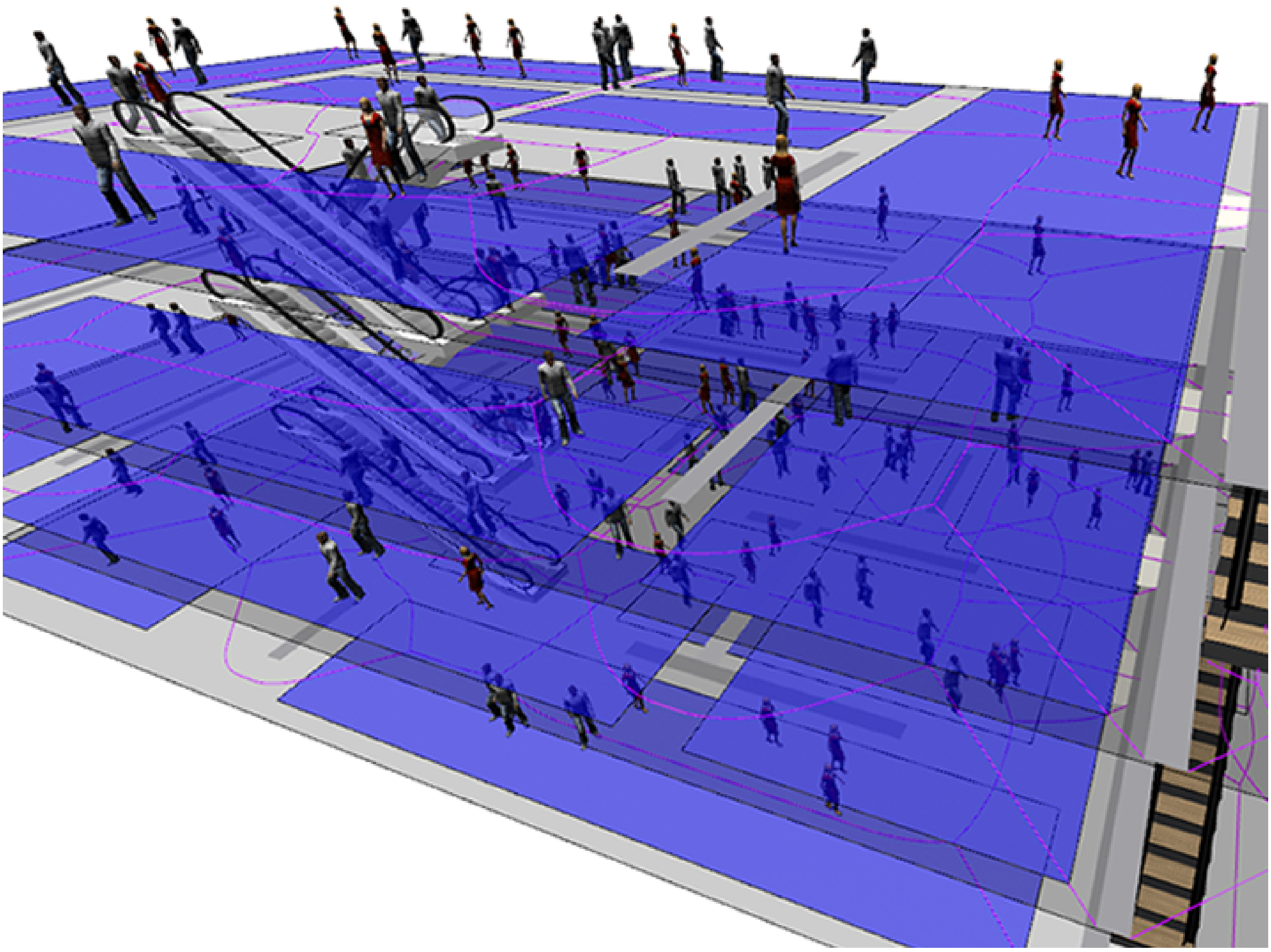


Figure 1: 3DModel view of the department store.

Exercise 1

Create a model of the first floor of the department store.

We will first draw the environment and set the Agent Input such that the department store is functioning as on a normal day.

Use a Height Layer of size 20 by 40 meters to represent the ground floor of the department store and some outside space. Use the snap to grid option or set specific coordinates for the Height Layer. In the selection menu of the 2D Builder layout you can enable and disable the snap to grid and also set the grid size. To set specific coordinates for an element switch to the Select mode and double click on the Height Layer, the dialog of the Height Layer will appear. Switch to the coordinates page to set the top left (TL) and (BR) coordinates. Change the name of the layer to Ground.

The store has different departments selling products including clothing, home appliances, toys, cosmetics, sporting goods and toiletries. We use a Commercial facility to model each of these departments. Furthermore, we use an Entry/Exit area to model the Entrance and use several Obstacles to model walls that separate some of the departments. Together these elements form the environment of the Ground floor of the department store, see Figure 2. Note that we use an Obstacle to divide the Height Layer into two parts. Later we will use the smaller area on the right for emergency fire escape stairs.

Switch to the Activities toolbar and draw an Entry/Exit area on the left side of the Height Layer to model the Entrance of the store. It is not a problem if the Entry/Exit area is (partially) drawn outside of the Height Layer. In that case make sure that you check the option *Ensure walkable area* of the Entry/Exit area. You can find this option on the Location page of any Activity Location. Next draw one Commercial facility which represents a single department. Before we draw the other departments we will first change the settings of this Commercial facility and then make copies of this element that will represent the other departments.

For this case we assume that customers that visit a department on average spend 5 minutes in this area. The time an agent spends at an Activity Location can be set in the dialog of the Activity Location. One of the important settings that determine the time an agent spends at a Location is the Activity time. This is the time in seconds an agent will spend at this location. Set the Activity time to 120 if you want an agent to stay in the location for exactly 2 minutes. The time does not have to be fixed, you can also use a probability distribution. If the Activity time is set to `NegExp(300)` each time an agent visits this location a random number is drawn from the negative exponential distribution with a mean of 300. On average agents will spend 5 minutes at this location. Note that there are other settings that can influence the time an agent will spend at an Activity Location, making the period longer than the Activity time. For this case we set the Activity time to `mins(NegExp(5))`. Switch to the Select mode and double click on the Commercial facility. The Activity time can be found on the General page.

By default agents are not shown when they enter a Commercial facility. To be able to see the agents when they enter such a location set the option *Exclude Agent* to No instead of the default option Exclude_Hide. The Exclude Agent setting can be found in the dialog of the Commercial facility on the Location page. Finally change the name of the Commercial facility to Department. After you have fully set up the first department you can make copies to quickly create the other departments. Select the Commercial facility and press Ctrl+C to create a copy, and Ctrl+V to paste it. The Commercial facilities are now draw on top of each other. Switch to the move mode and drag the copy to a different location in the Height Layer. Create six departments on the ground floor leaving some space in the middle for the escalators. Finally switch to the draw toolbar and draw the Obstacles.

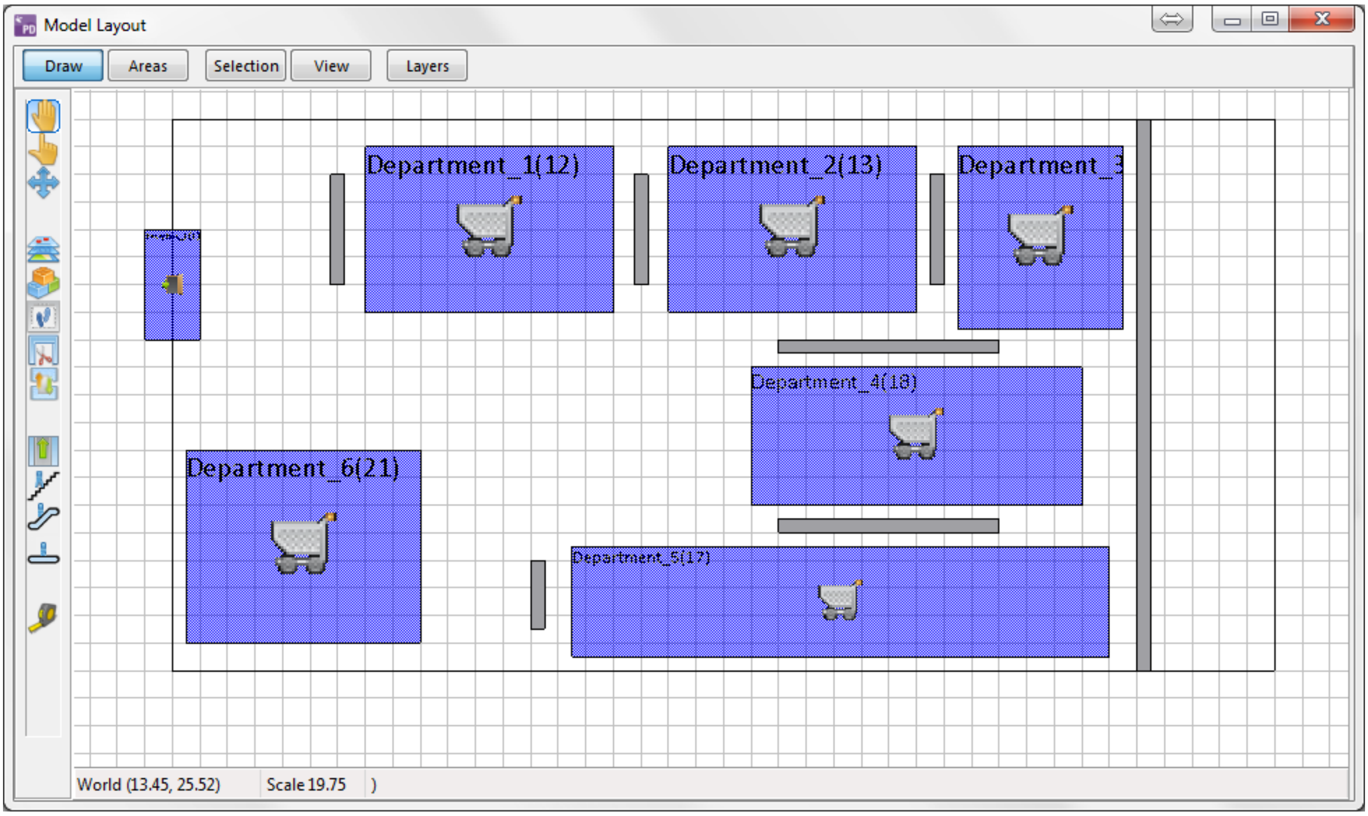


Figure 2: 2D Model view of the ground floor of the department store.

The next step is to define the behavior of the agents. In this case we choose to model customers arriving at the store, visit several departments and then leave the department store. Later we will add an element to start the evacuation. For evacuation analysis it is also possible to fill each floor with a certain number of customers and start the evacuation immediately.

Open the Agent Input dialog and switch to the Activity page. Since the Entrance is also the exit of the department store check the Revisit allowed option of the predefined Exit activity this setting can be found on the Location page of Agent Activity dialog. Next add a new activity named Shop. This Activity takes place at all Commercial facilities. Therefore set the Activity type to Commercial facility and the Activity group to ***ALL***.

We assume that a customer randomly chooses one department to visit. We can model this with an Empirical distribution. Switch to the Location page and set the Location distribution to Empirical and assign percentages to each of the departments such that they sum to 100%, see Figure 3. Switch to the Activity route page. Edit the Default_Route. Make sure the Agents first have the Entry activity, then the Shop and finally the Exit activity. Switch to the generator page.

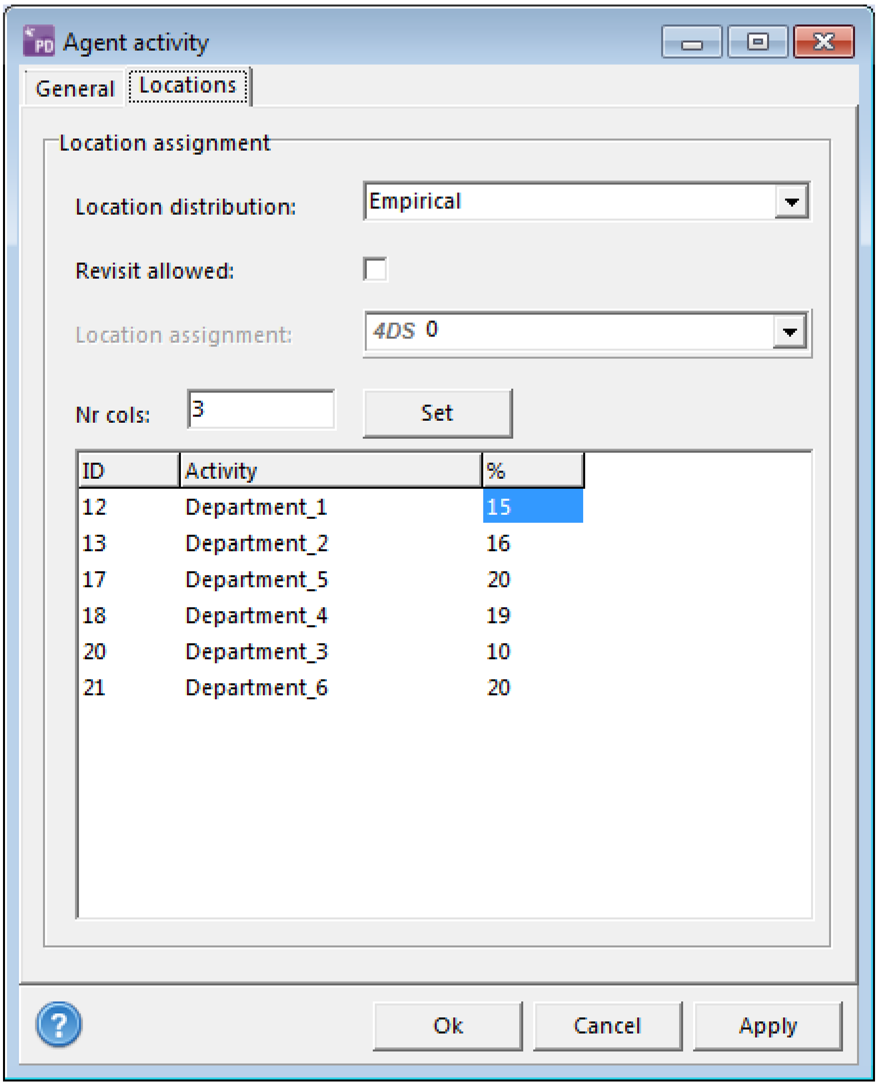


Figure 3: The Locations settings of the Shop activity. The location is assigned based on an Empirical distribution. In the table percentages for each location define the probability that an agent will visit this location when performing the shop activity.

The arrival process for visitors of a store can often be modeled as a Poisson process. This means that the time between two arriving customers is exponentially distributed. The generator can generate agents based on an arrival list, but it is also possible to configure it as a random generator. Edit the Default generator and apply the following settings; Set the Repetitive mode to continuous, set the Offset time to `NegExp(4)`, i.e., a negative exponential distribution with mean 4 seconds and set the delay time to 0. Make sure the arrival list has a single row and creates one agent at time zero, see Figure 4 and Figure 5.

Run the model to check its validity.

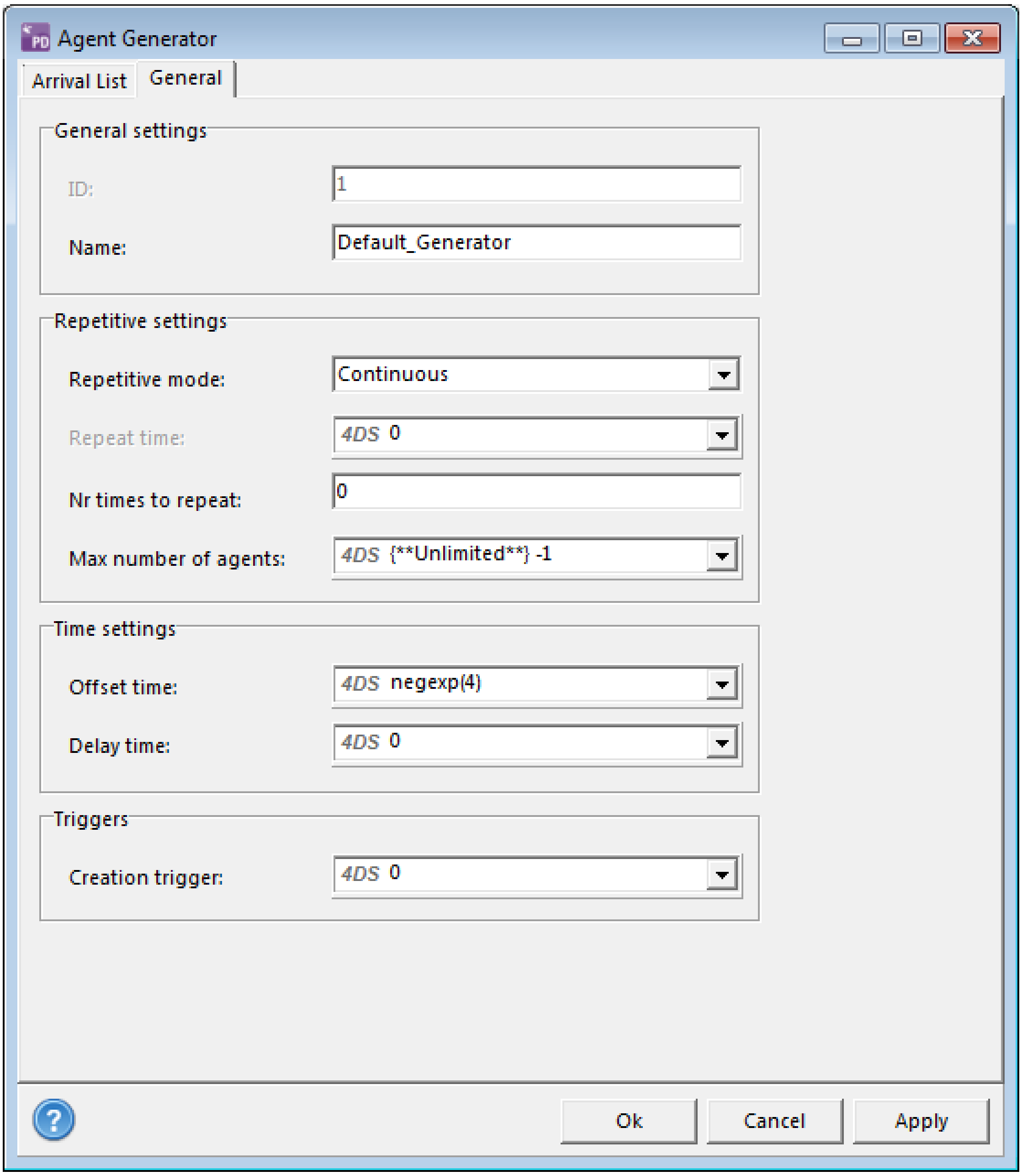
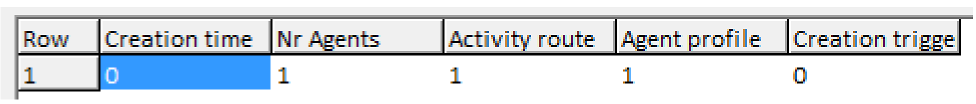


Figure 4: The General settings of the Agent generator configured as a random generator.



Row	Creation time	Nr Agents	Activity route	Agent profile	Creation trigge
1	0	1	1	1	0

Figure 5: The Arrival list settings of the Agent generator configured as a random generator.

In real life, most of the customers will of course visit more than one of the departments. Some customers visit several departments while others have a specific purchase in mind and visit only one.

Exercise 2

Make sure that most customers visit several departments, i.e., some will only visit one, others two or more with a maximum of five.

To make sure that visitors go to several departments we only need to adapt the Activity route. If we simply add five Shop activities to the route, every customer will visit five departments in a random order. To model customers who do not visit all five departments, we have to make some changes to the settings of the Shop activity. Essentially, we will make the visitors skip each Shop activity with a certain probability. One easy way to model this is to adapt the percentages in the Empirical distribution of the Shop activity, making the percentages no longer add up to 100%, see Figure 6.

If the probabilities do not add up to 100%, the remaining fraction means "skip this activity". If you reduce the probabilities so that they add up to, say, 60%, the visitors will on average visit 60% of the 5 departments, which means that on average they will visit 0.6*5=3 department stores. The probabilities for each number of visited department stores can be computed using a binomial distribution. The probability of getting exactly k successes in n trials is given by the probability mass function:

$$f(k;n,p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

for k = 0, 1, 2, ..., where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

This leads to the following probabilities that a visitor will visit a number of department stores.

# Visited	Probability
0	1.02%
1	7.68%
2	32.04%
3	34.56%
4	25.92%
5	7.78%

- The probability that a department store is visited by a visitor is 60%. This probability is the same for all department stores. So when there are 100 visitors each department store will on average get 100 * 0.6 = 60 visitors.
- The is no correlation between the visits; a shopper who visits store 1 is neither more nor less likely to visit store 2. To model correlations you can define multiple routes where each route contain a specific number of department store in a specific order. Alternatively you can write your own distribution via the user defined, where you take the currently visited store into account. You can use labels for this or use the functions Agent_GetVisitedDestinations and Agent_GetVisitedDestination.

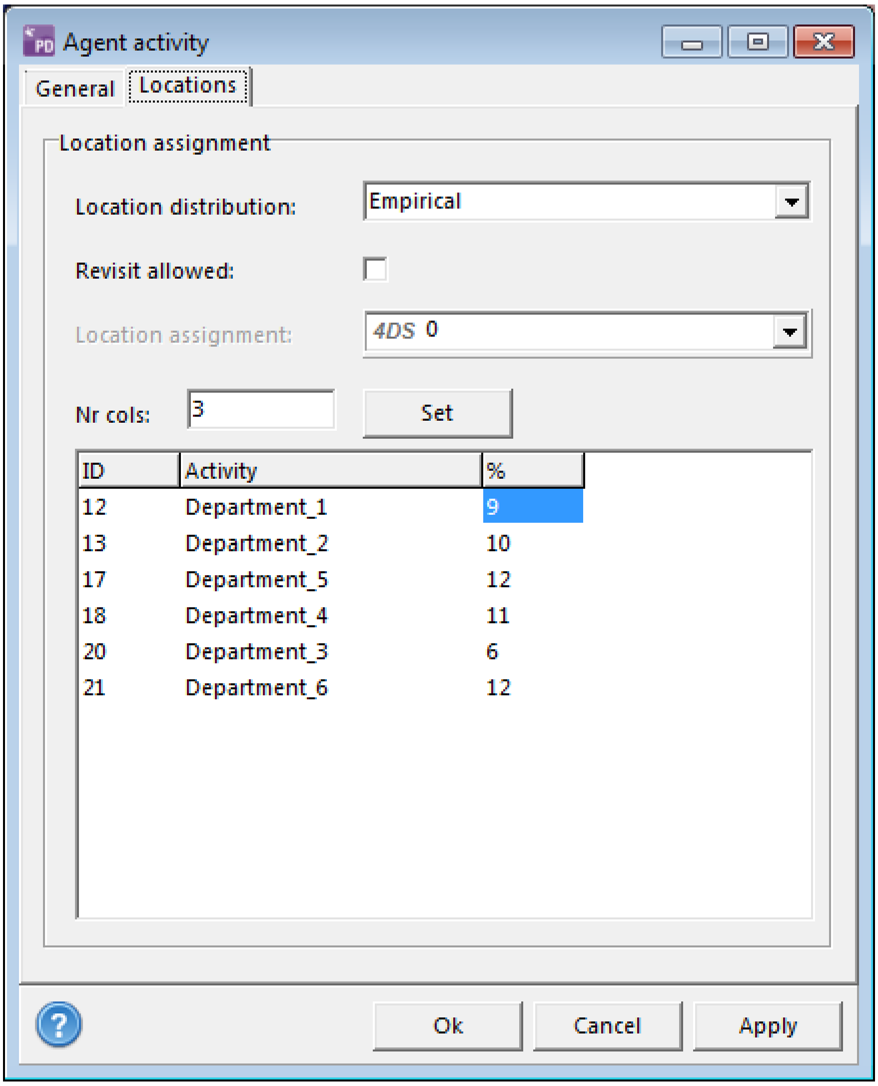


Figure 6: The Locations settings of the Shop activity. Note that the percentages of the Empirical distribution do not add up to 100% the remaining fraction gives the probability that the agent will skip this activity.

Before we add the First, Second and Third floor to the environment of the department store we will first create an evacuation model.

4.1 The Action timer element

It is very easy to create an evacuation model in Pedestrian Dynamics. After you have built the environment and added behavior to the Agents the only thing you have to do is add an Action timer element and apply the correct settings. An Action timer is a trigger, it allows you to determine the time that an incident occurs and evacuation has to start; of course this can also be based on a probability distribution. It is also easy to model what happens after the incident. Settings of the Action timer allows you to determine what happens to agents that still want to enter the building and when and how existing agents react. Will they leave the building immediately or will it take some time before they start walking to the exit? Furthermore, you can make changes to the infrastructure: an emergency exit should only be available during an emergency. The Action timer also displays a clock which can be reset when the incident starts counting the time that elapses until all agents have left the building.

Exercise 3

Include an Action timer in your model to cause a trigger at which point the pedestrians will leave the building.

The Action timer can be found on the Areas toolbar. Select the Action timer icon  and draw it somewhere in the model, see Figure 7. In this case the location of the Action timer will have no effect.

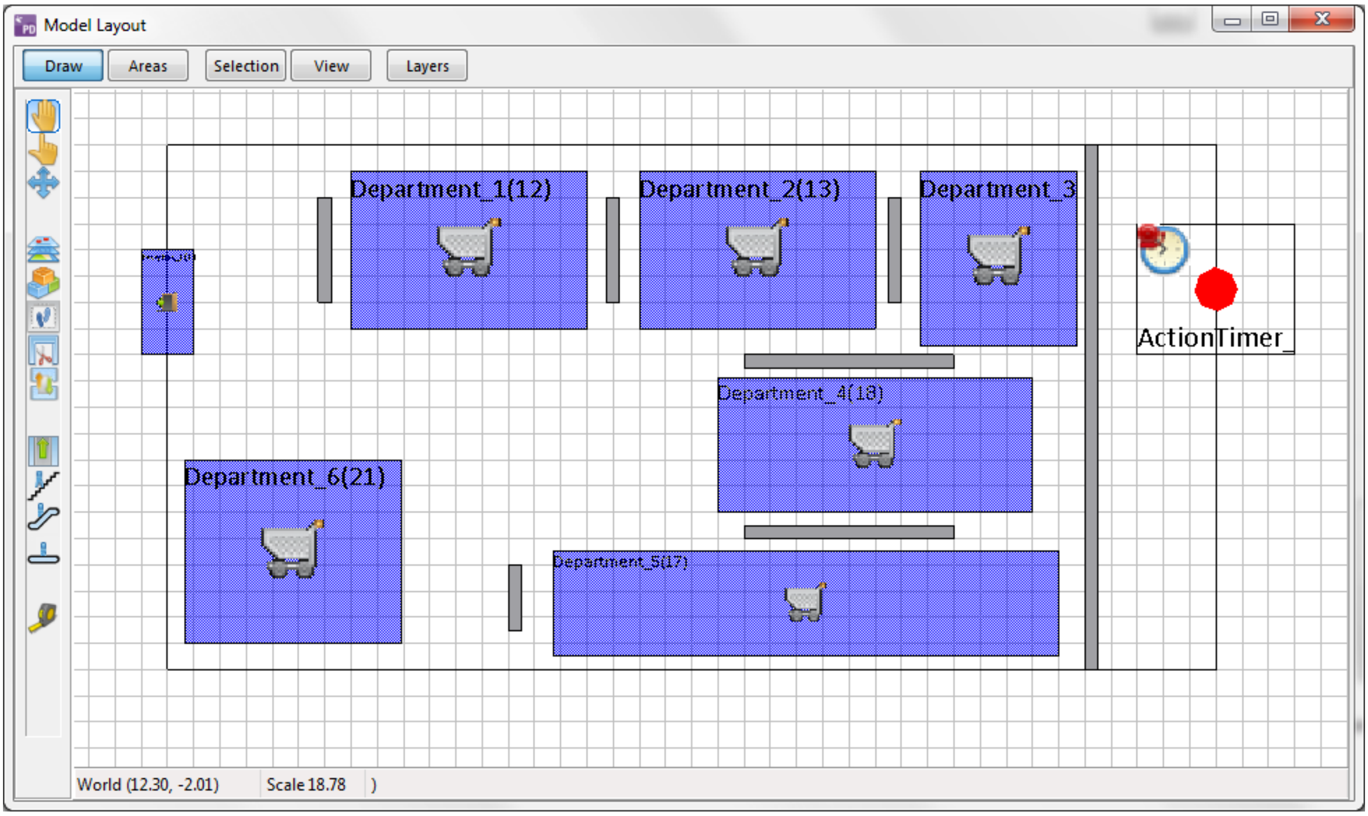


Figure 7: The Model layout with the ground floor of the department store. An Action timer is drawn on the right side of the ground floor.

The Action timer allows us to model the occurrence of an incident and the effect it has on the agents and the infrastructure. In this case we use the Action timer to model the time that the alarm system of the department store goes off and moment that the evacuation starts.

Check that after you have created an Action timer in your model, a new predefined Activity named Emergency_Exit is added to the activities and an Activity route named Emergency_route is added to the Activity routes. By default the agents do not yet use this activity and route. We want to change the settings of the Action timer such that after the alarm system goes off agents use this route to find an exit. Double click on the Action timer, the dialog of the Action timer will appear, see Figure 8.

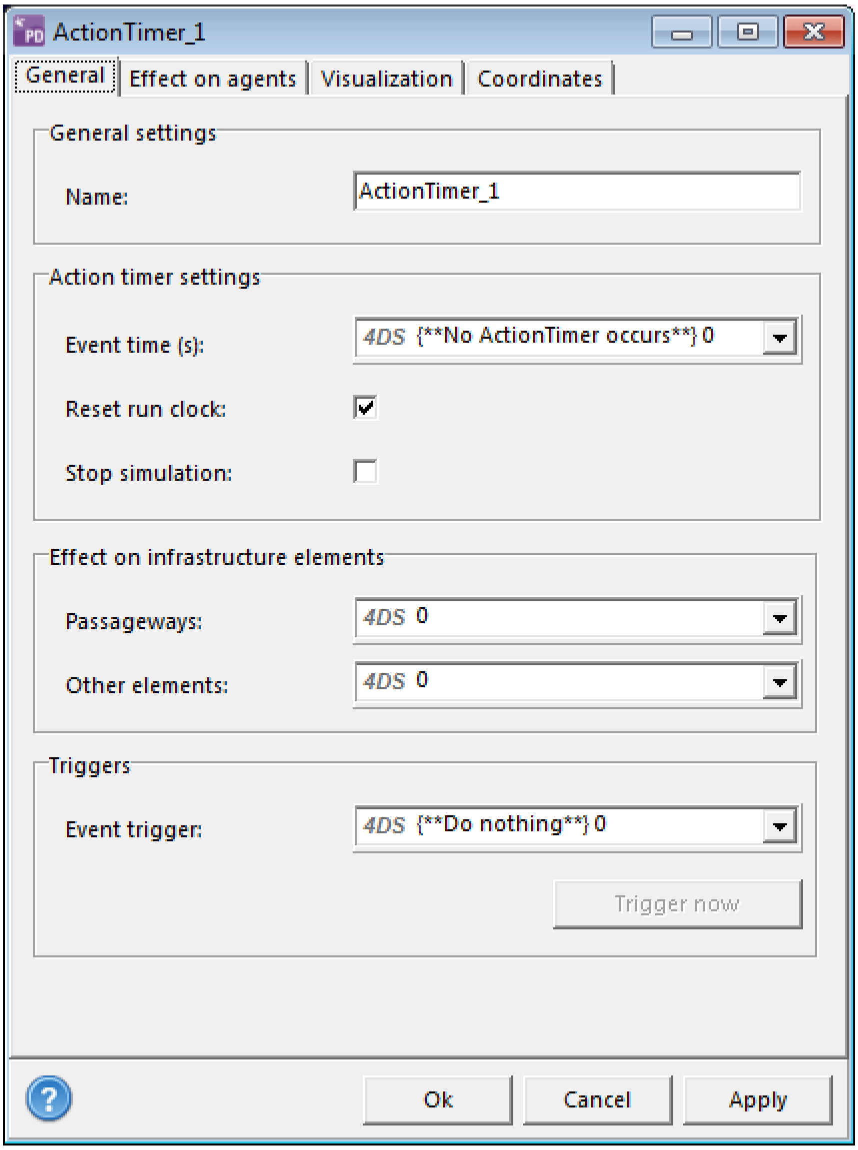


Figure 8: The dialog of the Action timer.

The first important setting is the Event time, this is the time at which the incident occurs. If the *Reset run clock* option is checked, the clock visible in the Model Layout is reset and turns red at the moment the event occurs counting the time from the start of the incident. The simulation clock is not reset and will keep running counting the time from the start of the simulation. In this case we choose that the incident will occur after half an hour. At that moment the department store is already filled with agents. Set the Event time to 1800 seconds. Note that you can also use the 4DScript function hr and type `hr(0.5)`. Make sure that the option Reset run clock is checked.

All changes in behavior of the agents due to the incident can be set on the Effect on Agents page of the Action timer, see Figure 9.

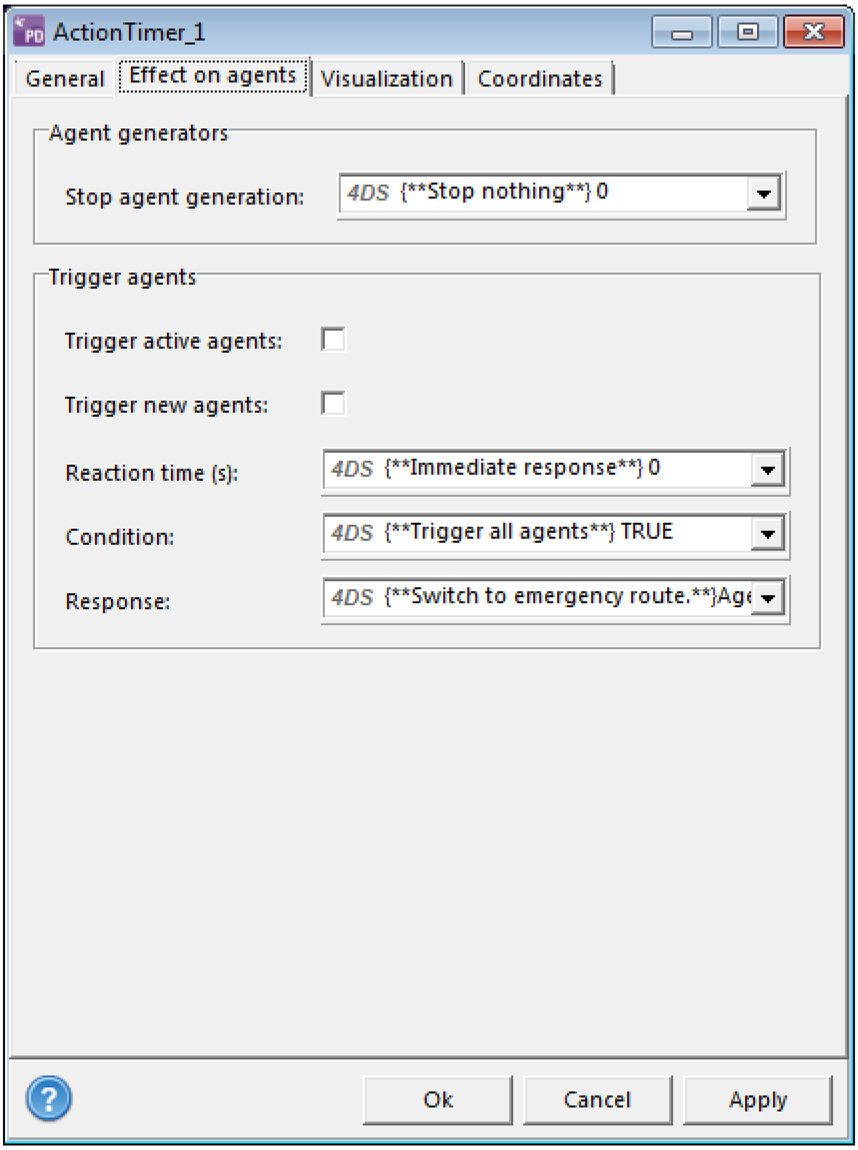


Figure 9: The Effect on Agents page of the Action timer dialog.

In this case we assume that the moment the alarm of the department store goes off no new agents enter the department store. Therefore we need to stop the agent generation. Switch to the *Effect on agent* page of the dialog of the Action timer. Select the predefined logic {**Stop all immediately**} for the *Stop agent generation* property, this logic will make sure that all generators will immediately stop generating any agents. Other logics would allow you to only stop specific generators or stop them after a certain time has elapsed. Maybe a generator is responsible for the generation of agents out of a train. The train that just entered the station should still generate the agents from that train.

The agents within the building are all following an Activity route. The moment an alarm goes off they should stop with their planned activities and need to find a way to leave the building. For this case we assume that all the visitors in the department store react immediately and try to find the exit. We can create this behavior with the setting available in the *Trigger Agents* group on the *Effect on agents* page of the Action timer. Check the *Trigger active Agents* box. This will make sure that all the agents that are in the model the moment the event is triggered will get the defined Response after a specified *Reaction time* if the agent fulfills the *Condition*. Set the *Reaction time* to 0 {**Immediate response**} such that the agents will react immediately after the event occurred. In practice for instance a fire bell is often ignored. It can take some time before people react and start to move to the exit. This can be modeled with the reaction time. The *Condition* can be used to select a certain group for the response. Maybe only a single floor needs to be evacuated so the response is only for the agents on a specific Height Layer. In this case we want all agents to evacuate so we can set the condition to True {**No specific condition**}. Most important is of course the Response itself which we can set to the Logic {**Switch to emergency route**}. By default this logic will make sure that the agents get the Emergency_route as their new route. By default the agents will select an Exit they can reach with the LeastEffort. The Action timer settings should be as in Figure 10.

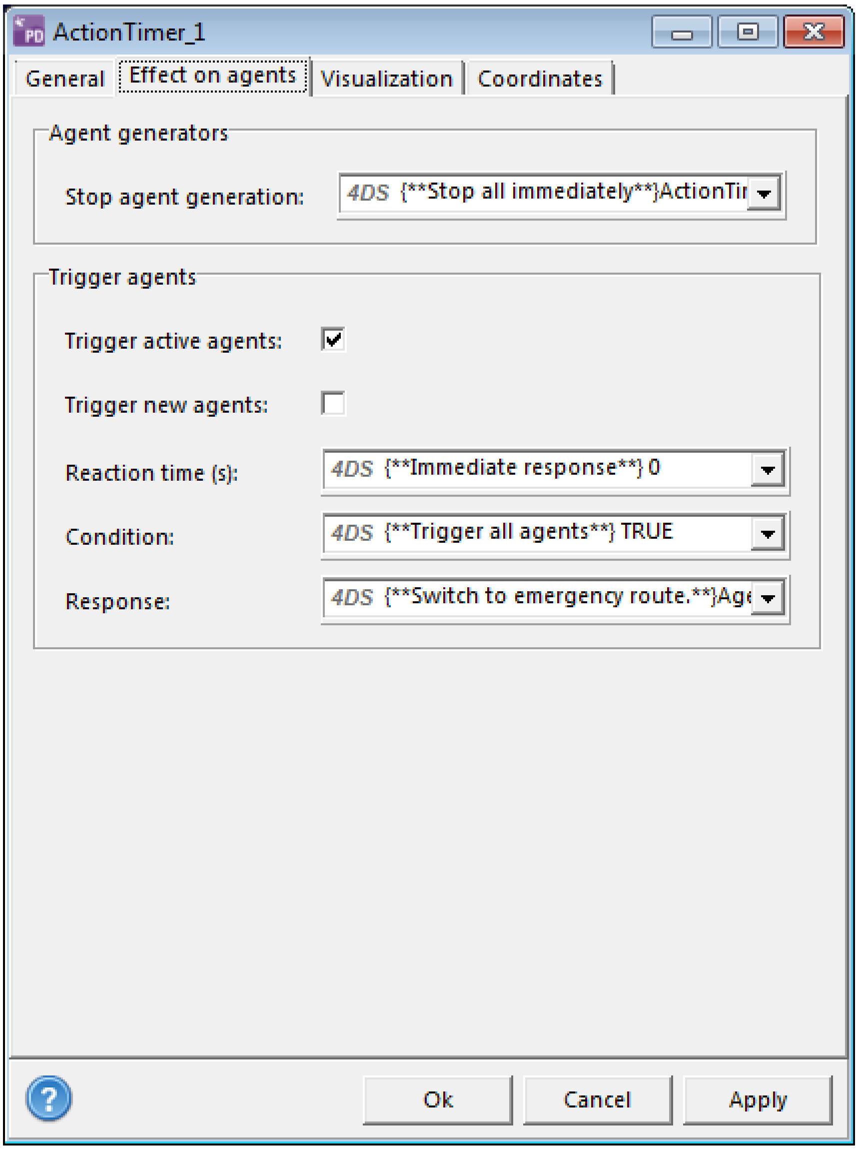


Figure 10: The Effect on agent page of the Action timer. After the Incident stop the generation of all agents and reroute the agents to the Emergency_route.

Finally switch to the general page of the Action timer and make sure the option Stop simulation is checked. This will make sure the simulation stops the moment the incident occurs. This allows you the set the speed of the simulation to unlimited until the incident occurs after that you can set the Run Control to *Slide control* and decrease the speed. Start the simulation again pressing Run on the Run Control.

Run the model, watch how the agents leave the building and check the evacuation time.

4.2 Using a Passageway element as an emergency exit

Larger buildings will often have emergency exits. These exists are not to be used during normal operation. A Passageway element can be used to represent an emergency exit. A Passageway can be found on the draw toolbar. A Passageway is an area where agents can walk and if it overlaps an obstacle then the intersection also becomes an area where agents can walk. A Passageway can represent a door in a wall but it can also represent a turnstile. It is possible to control the direction that agents can pass the passageway. You can set it to uni-, bi-directional or even unavailable. We will use these features to represent an emergency exit which is only available during an emergency.

Exercise 4

Create an emergency exit in the environment.

Switch to the draw toolbar, select the Passageway icon and draw a rectangle shaped area overlapping the obstacle that represents the wall of the department store, the result should look as Figure 11. We wish to control the direction of the flow of the passageway, i.e. bidirectional during the emergency and else unavailable. The green arrow shows the direction in which the agents can pass.

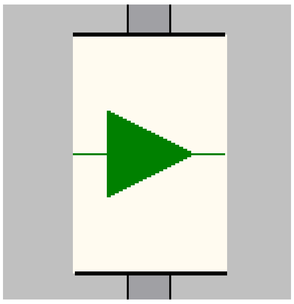


Figure 11: A Passageway element.

There are in total four direction settings for the passageway:

- Unavailable, i.e., agents cannot pass the passageway
- Up, i.e., only agents coming from the up direction can pass
- Down, i.e., only agents coming from the down direction can pass
- Bidirectional, i.e., agents coming from both directions can pass

Note that Up in this case means from left to right and Down from right to left. Rotate the passageway if you want to form a gateway in the other direction (North-South).

In this case we want the passageway to be unavailable, because during the first time period of our simulation there is no emergency situation and agents are not allowed to use it. Open the dialog of the passageway and set the direction to Unavailable. Also set the discomfort factor to zero. This factor is used when agents plan their path, the higher the value, the less attractive the passageway becomes. During an emergency agents will not hesitate to take the emergency exit.

To make sure that agents will use the emergency exit we need to create another Entry/Exit area outside the department, see Figure 12. Draw an Entry/Exit area on the bottom right keeping some space for emergency stairs.

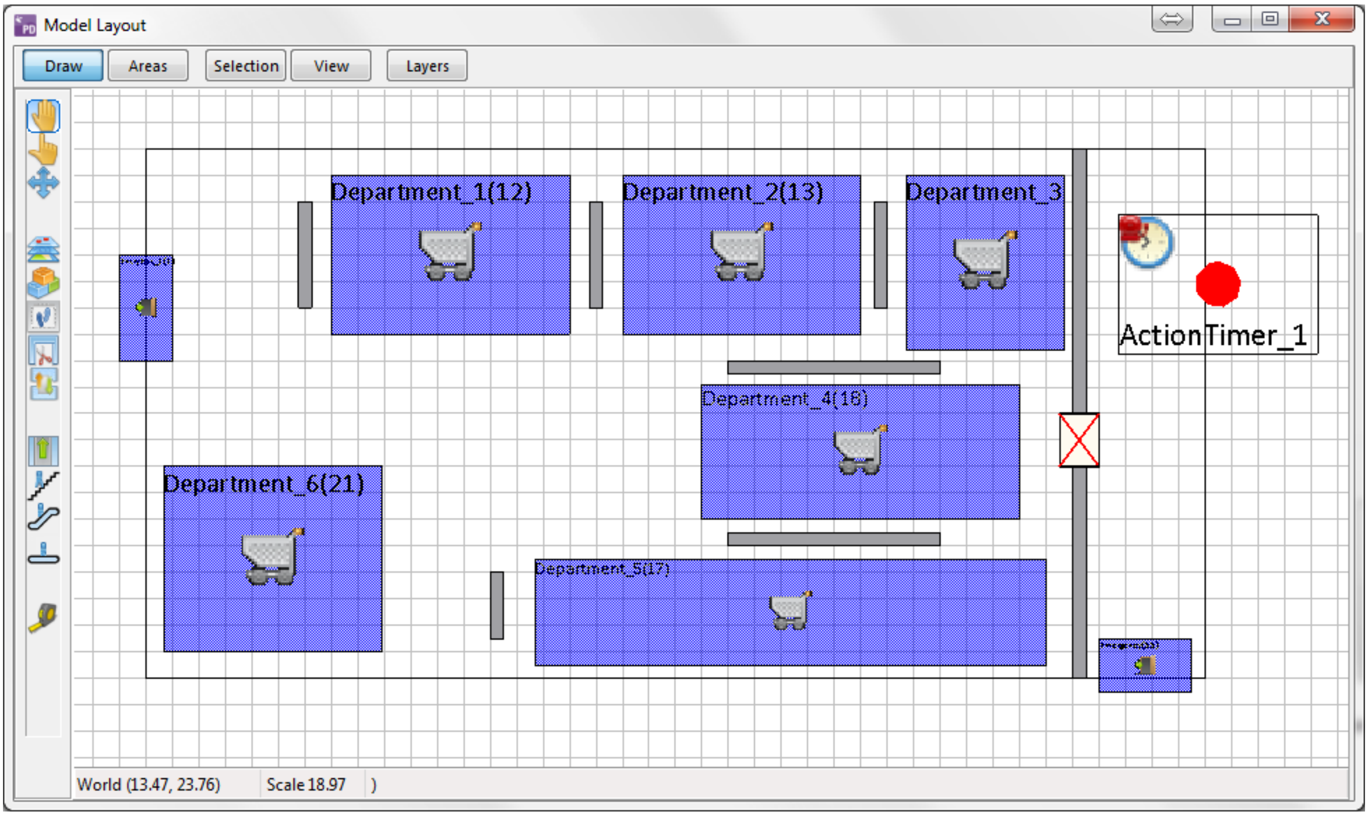


Figure 12: The Model Layout with the ground floor of the department store with an Action timer. The Passageway used as an emergency exit is closed before the incident occurs.

We also need to change the Entry and Exit activities making sure that only the Entrance is used to generate agents and that during the first time period agents only leave through this Entry/Exit area. Open the Agent Input window and switch to the Activities page. Edit the Entry activity. Set the Location distribution to Empirical and set the percentage of the Entrance Entry/Exit area to 100. Do the same for the Exit activity.

Finally, we need to make sure that our Action timer changes the direction of the emergency exit from unavailable to bidirectional. Open the dialog of the Action timer. Set the Passageways field to the {**Update all**} logic. Double click on the logic and make sure that the second parameter of the function is set to DIRECTION_BIDIRECTIONAL.

Run the model, watch how the agents leave the building also through the emergency exit and check the evacuation time.

4.3 Using an Escalator in an emergency model

In the event of an emergency modern escalators can stop smooth and controlled. After the stop pedestrians can still use the escalator to walk up or down. It is very easy to model this using the Action timer.

Exercise 5

Add a new floor to the department store and create escalators to connect the floors. Also create emergency stairs outside of the building.

It is very easy to create multiple floors with the same layout. Open the Height layer settings panel if it is not yet visible on the right hand side of the Model Layout. To open the panel click Height layers in the top menu of the Model Layout. Note that we only have a single Height layer named Ground. Select the ground floor Height layer, switch to the select mode and click the Height layer. At the bottom of the Model Layout window you can find the World coordinates, the scale but also the name of the element that is selected. Check that you have selected the Height layer. Press Ctrl+C to make a copy of the Height layer with all its elements. Press Ctrl+V to paste it. Note that a layer named Copy of Ground appears in the Height layer settings panel.

Click on the eye icon before the Height layer named Ground. *Copy of Ground* is now the only visible Height layer. Remove the Entry/Exit areas and the Action timer. Open the dialog of the Height layer and change the name to

First and set the z-Loc to 4 meter, the result should be similar to Figure 13. If you like you can move and resize the Commercial facilities and Obstacles to create a different layout for the First floor. If you want to add some Obstacles or Commercial facilities do not forget the make the First floor layer the Active layer. To make a layer the Active layer, click on its name in the Height layer settings panel. A blue bar indicates the active layer.

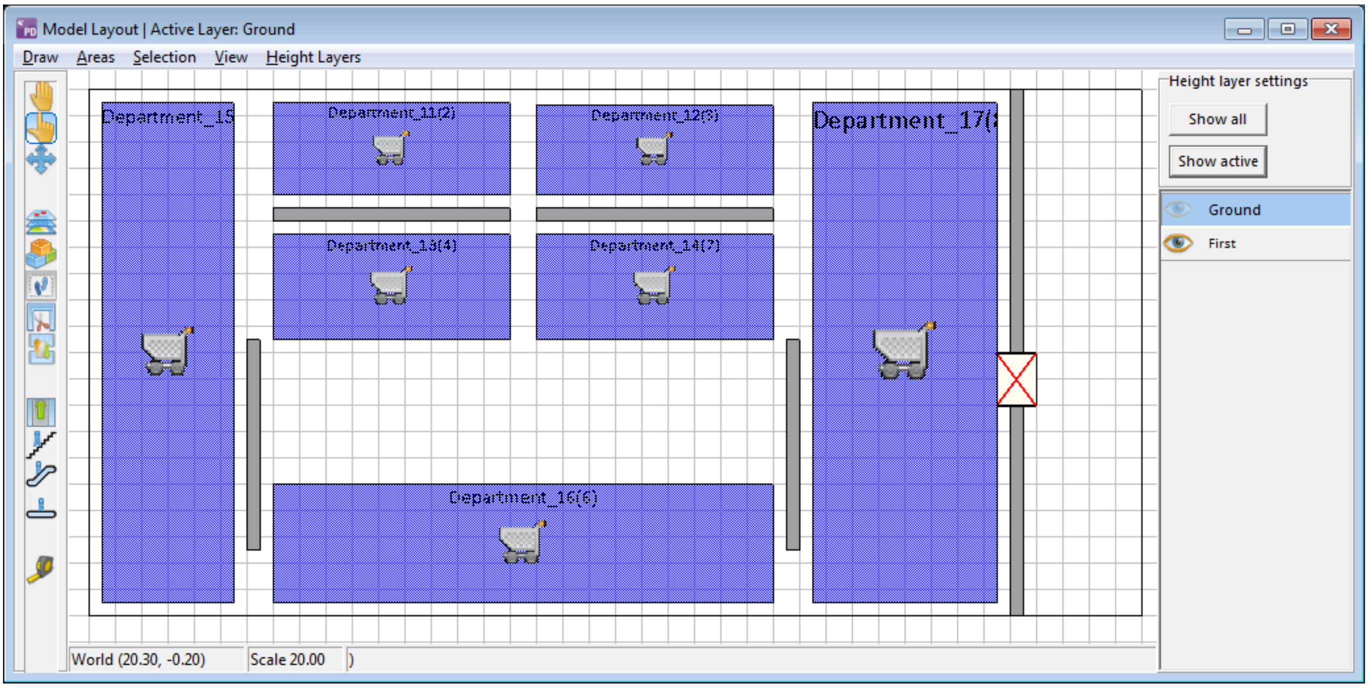


Figure 13: The Model layout with the first floor Height layer visible. The Ground floor is still the active layer.

We still need to form a connection between the two Height layers, therefore we will use an escalator element. Before we draw the escalator, make the ground floor the only visible Height layer and make it the Active layer. Draw two escalator elements above each other. We want to make sure that the escalators are configured crisscross, i.e., when we later add escalators to the other floors all escalators that go in one direction are stacked, see Figure 14. Note that there are two sloping lines that intersect at the end of the escalator. The side where the lines are separate is the side that is connected to the lower level. Select one of the escalators and press F7, rotate the escalator 180 degrees. The green arrows give the direction of the elevator, i.e., is it moving Up or Down. Open the dialog of the escalator. Check the Transfer box and set the "Layer from" and "Layer to" fields of both escalators to Ground and First, respectively. Set the direction of the rotated escalator to Down.

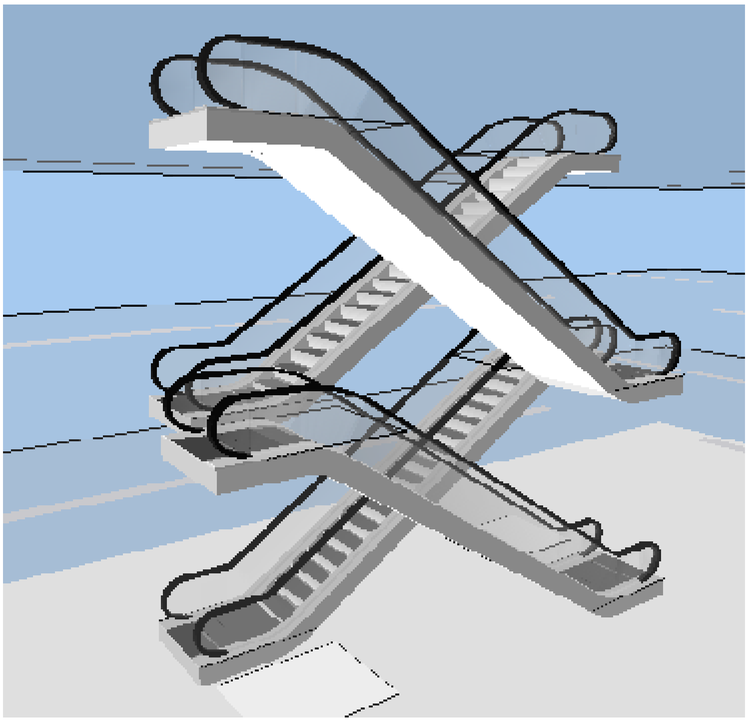


Figure 14: Criss Cross configuration of escalators.

Draw a stair element on the outside of the department store. Rotate it 270 degrees and give it proper length. Open the dialog of the stairs and set the *Layer from* to Ground and the *Layer to* to First. The result should look similar to Figure 15.

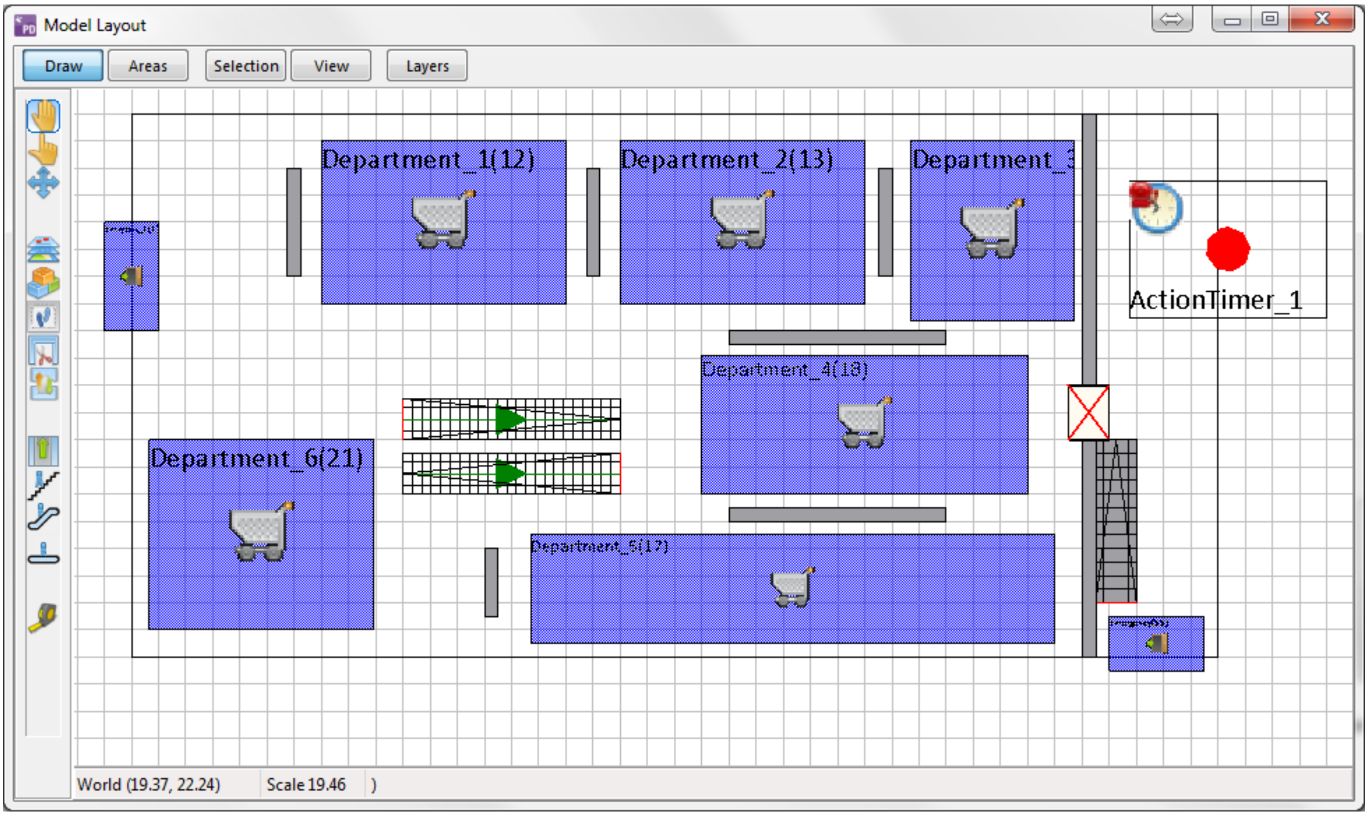


Figure 15: The Model Layout with the first floor Height layer visible.

The final change we have to make to our model is the Location distribution setting of the Shop activity. The *Location distribution* of the Shop activity was set to Empirical distribution. The new Commercial facilities can be found in the table but their percentages are all zero. Open the Agent Input dialog and edit the shop activity. Change the percentages such that all Commercial facilities have non zero percentage.

Run the model and check that the agents visit the First floor. Note that after the incident the escalators still work and the agents only use the escalator moving down.

We need to make a change to the Action timer if we want the escalators to stop running and allow agents to walk on them. Open the dialog of the Action timer. In the "*Effect on infrastructure*" group, set the Other elements field to the {**Update All**} logic. This field applies to escalators, stairs and moving walk elements. Double click the field and make sure that the direction is set to bidirectional. Stairs are typically always bidirectional and after an emergency stop, escalators are bidirectional too.

Exercise 6

Add two more floors to the department store and create escalators to connect the floors. Also create emergency stairs outside of the building.

We need to repeat the steps for exercise 5. Make two copies of the First floor. Change the name and the z-Loc of the new Height layers. Add escalator and stair elements to connect the floors. Change the percentages of the Location distribution of the Shop activity such that all locations have a non zero percentage.

Run the model and determine the evacuation time.

5 Output analysis

After you have created a model with Pedestrian Dynamics, having drawn the environment and set the required agent behavior, you can run an experiment to simulate the pedestrian flow through the environment. After the run you can examine the output in many ways without having to run the simulation again. You can perform all kinds of measurements concerning the walking times and pedestrian densities that occur in the environment and easily create a report to present the outcomes.

In this tutorial you will learn the basics of examining the output with Pedestrian Dynamics. You will also learn how to run an experiment and how to measure the results of the experiment from the 2D output window.

5.1 Station hall construction work

Due to necessary construction work, part of the main hall of a regional train station needs to be closed off. The management is concerned with the safety and comfort of the passengers and wants to ensure that their norms for crowd densities will not be exceeded during the renovation.

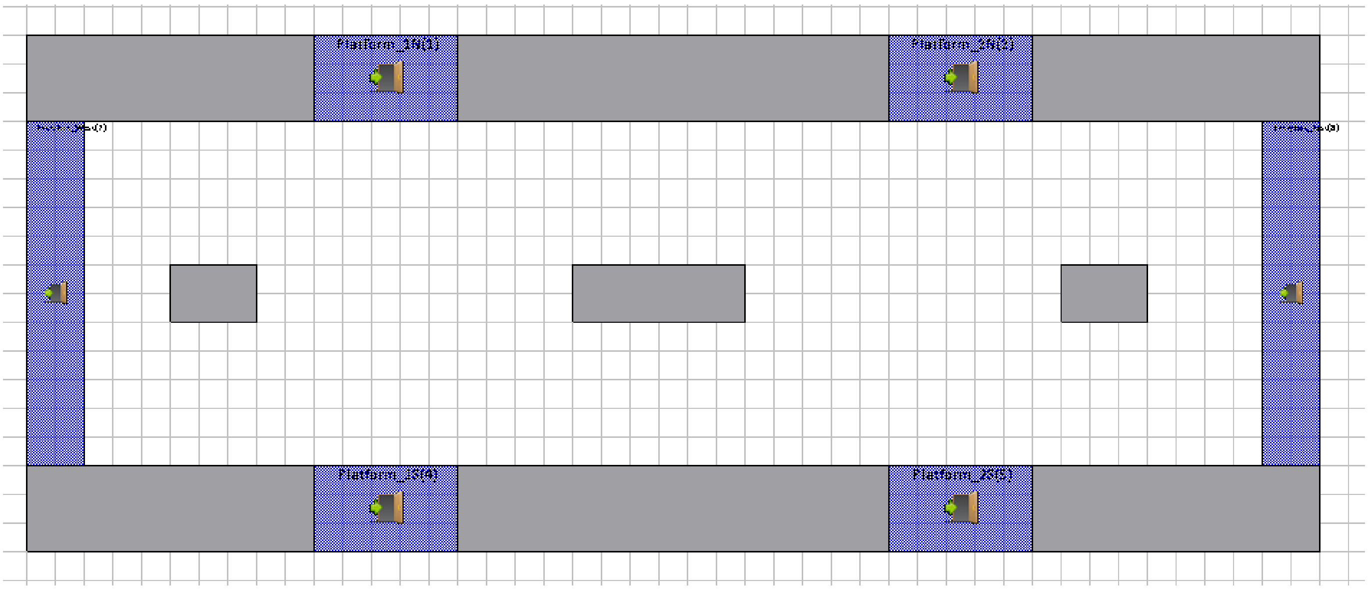


Figure 1: An overview of the layout of the main hall of the station.

The main hall is an east-west passage located below the two platforms, which run north-south. Figure 1 gives an overview of the station. The city center is located to the west entrance of the station, which sees considerably more traffic than the east entrance: 70% versus 30%. The stations time table repeats every 30 minutes according to the following schedule:

Time	Platform	# boarding	# disembarking	transfer (%)
X:06 and X:36	1	250 - 300	200 - 250	40%
X:10 and X:40	2	250 - 300	200 - 250	40%
X:22 and X:52	1	250 - 300	200 - 250	40%
X:25 and X:55	2	250 - 300	200 - 250	40%

Table 1: Hourly time table of the station, with traffic estimates.

You can see that each half hour two trains arrive at Platform 1 and two at Platform 2. In the table you see an estimate of the number of passengers that board and disembark each train during the morning rush hour. The table also gives the percentage of passengers that transfer to the other platform. Passengers who transfer to a train on the same platform do not appear in this table at all because they are not visible in the station hall. The station also functions as a East - West pass-through for about 1200 pedestrian per hour, during the morning rush hour 70% enters West and leaves East.

During the renovation work, part of the main hall of the station is closed off, creating a corridor with a width of 5 meter between the platforms. This means that all traffic needs to pass through this corridor except traffic between the West entrance and Platform 1 and traffic between the East entrance and Platform 2 as depicted in Figure 2.

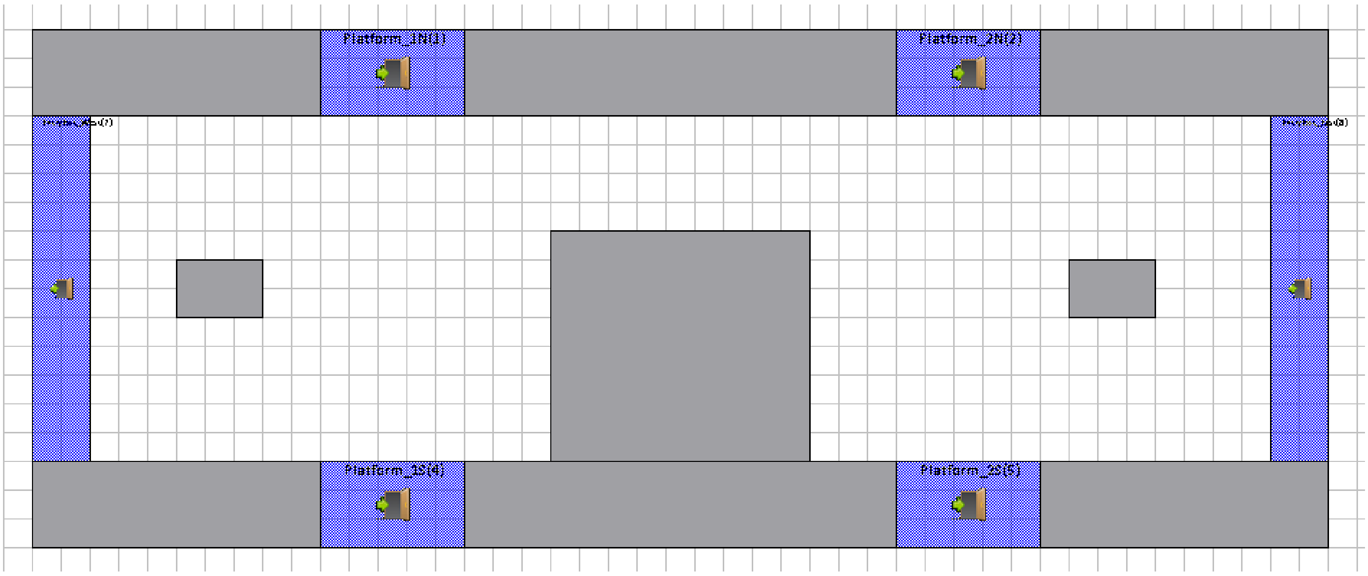


Figure 2: Station layout with construction work.

Exercise 1

> Using the data from Table 1, calculate the expected number of pedestrians crossing the corridor East-West and West-East per hour. Also calculate how many pedestrians enter and leave the West entrance of the station. We will use this later to validate the model.

Description of the model

The situation at the train station has been modeled in the file *station hall construction.mod*. Open this model in Pedestrian Dynamics and run it a few times to get familiar with it. Note that the passers, boarders, disembarkers and transfers have different colors: green, blue, red and purple, respectively. In the minutes running up to the arrival of a train, more and more boarders arrive. Then, when the train has arrived there is a large peak in traffic dominated by disembarkers.

The first half hour of the run is not yet representative because for technical reasons, the boarders only appear later. During the first few minutes, only passers enter the model. After six minutes the first train arrives, depositing a number of disembarkers and transfers. At this moment the model also generates the agents that will eventually board the train that arrives half an hour later on that platform. These agents are delayed in such a way that you get a realistic arrival pattern, i.e. most of the passengers arrive in the minutes before the train departs. The same holds for the other trains. You have to keep this in mind when analyzing the results from this model.

During a run you can click Agent Statistics in the Simulate tab of the main menu to get a breakdown of the number of agents in the model and per height layer. You can see how many agents are walking to their next activity and how many are busy with an activity. The window is updated dynamically. You can keep it open while the model is running to keep an eye on the number of agents. In this specific model there are no activities in which the agents spend any time so you will only see a zero for that.

Exercise 2

> Run the model and get an indication of the number of agents in the station at the busiest moments. What is the main cause for this peak in traffic?

5.2 The Experiment Wizard

Most of the other analysis tools can only be applied if you have recorded data during a run. This can easily be done in the Experiment Wizard. The Experiment Wizard contains facilities to automatically run the model several times, optionally with different parameters. For example, you can schedule five runs with regular expected traffic, and five others with 25% more traffic, then compare the results. For each scenario multiple simulation runs are required, we call these Replications. This is needed to ensure statistical significance of the simulation results. You should never base your conclusions on the results of a single run of a model, so for each parameter combination you should run multiple replications, then compare the results.

Exercise 3

Set up an experiment using the Experiment Wizard. Create one scenario which runs for 1.5 hours with 4 replications. After that start the experiment.

Click the Wizard button on the Simulate tab of the Main Menu. The Wizard opens with an introduction page. If you are going to create many scenarios with similar settings you can use the Check Defaults button to modify the initial settings of a newly created scenario. In this case we will only create one scenario. Click Next to go to the second page. One scenario has already been defined. Click Edit to modify it. Fill in a name and description of the scenario. You can see that the output will be written to a directory named PD_Results. Click Edit again to modify the important parameters of the scenario such as the run length and the number of replications. Click Ok twice to get back to the main experiment page. Click Next to go to the third and final page of the Experiment Wizard.

In this page you can determine which data needs to be recorded and start an experiment. For this experiment we need both the Output and the Footsteps to be logged so we can make all types of output graphs. If you would now click the Finish button, the Wizard would close but the settings will be saved with the model. If you open the Wizard again it will open on this third page so you can immediately start an Experiment.

Click Start Experiment. The Builder window will be closed and you will notice that the clock is running. Occasionally, a message will appear that PD is writing results to file. After a while, a message appears that the experiment is finished. Switch to the Results tab of the Main Menu. Here you can open a special 2D Output window which is specialized to analyze the results of an experiment run. Normally, the data from the last run is already loaded unless you disabled this option on the third page of the Experiment Wizard.

To open data from another replication, click the Open button. A window will appear in which you can select a replication directory. The experiment data is organized as follows: by default all data saved in the PD_Results directory. In PD_Results, each scenario has its own subdirectory. This means that it is very important to give scenarios a unique name, otherwise data will be overwritten. Within the scenario directory, each replication gets its own numbered folder. For example, the second replication of experiment *Experiment1* will be stored in *PD_Results\Experiment1\Rep2*. Select the replication you want to view and click Load.

The 2D Output window always shows the results from a single replication. To compare multiple replications you have to open them one by one. Later in this tutorial you will learn how to set up a report which can then be created for each replication.

5.3 The Result player

After an experiment you can use the Result Player to play back the events from the selected replication. Click the Result Player icon on the Results tab of the main menu and wait for the Player controls to appear, see Figure 3.

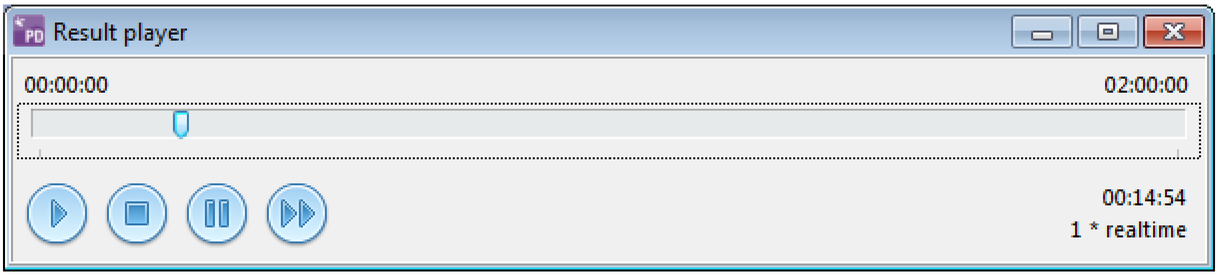


Figure 3: Result Player controls.

You can use the player controls in an intuitive way to control replay of the events of the run. The results can be viewed not only in the 2D Output window but also in any of the other display windows, such as the 2D Builder and the 3D Viewer. The rightmost >> button can be used to change the replay speed, it cycles between 1x, 2x, 3x, 5x, 13x, 100x and 0x the real time speed. You can also drag the time bar to move forward and backward in time. Sometimes it takes a few seconds for the animation to continue when you let go of the time bar; this is normal.

5.4 Frequency and density maps

Two very informative and easy to obtain analyses are the Density- and Frequency maps. Both use a color overlay to indicate which locations were crowded during this run, and which were not. Busy locations are shown red and purple, infrequently locations are blue.

The Frequency map is purely a counter: it counts the total number of people who have crossed that location during the complete run, or during the time interval you're analyzing. As you can see in the Figure 4, the colors indicate absolute numbers of people: the floor starts out white, it turns blue when people walk over it, then green, etc. This way, the Frequency map gives an indication of the utilization of the area.

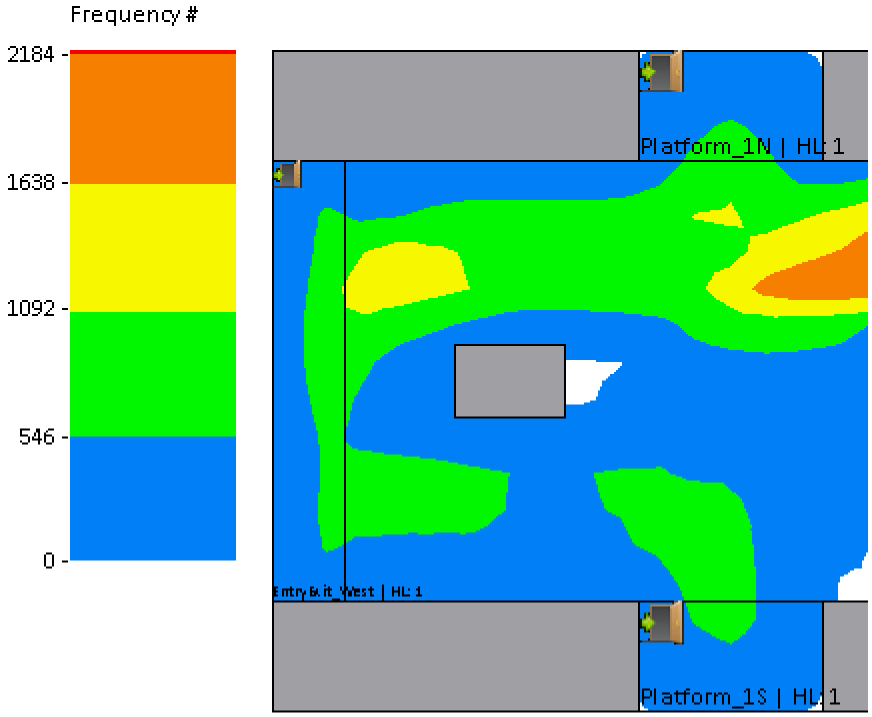


Figure 4: Frequency map of the West side of the station with colors indicating the absolute number of pedestrians passing there.

Where the Frequency map simply counts footsteps, the Density maps keeps track of the maximum density encountered over the run of the model. If a position was very crowded at any moment during the run of the model, the Density map will display it in red or purple. Essentially, the Density map breaks down time into a sequence of short time intervals, computes the average density during each time interval and shows the highest density it finds. The Density map is very useful as a quick indicator of whether there are bottleneck areas or choke points.

It is important to realize that the Density map shows the highest density that occurred. You should be careful to interpret this correctly. First, the map does not show you how long a high-density condition persisted. A location will be drawn in purple if a high density was ever recorded there, regardless of whether it lasted for 20 seconds or 2 hours. To examine whether this is the case, see the explanation of the Area Density Graph below. Second, the densities shown in the Density map are highly sensitive to the granularity settings. If the time intervals are very fine-grained, it is much more likely for a spike to occur than if the data is averaged out over fewer, coarser time intervals.

By default, the Frequency- and Density maps take their data from the full run. The same holds for the various other tools discussed below. There are however many situations where it is better to focus on a smaller time interval. For example, it is usually better to exclude initial minutes of the run because the model starts empty and the data is not yet representative for a steady-state model. Also, in models with time tables, it may be interesting to analyze separately what happens at certain times of day.

Exercise 4

> Use the Frequency- and Density maps to determine the critical areas in the station hall. Compare data from multiple runs. Describe what you see.

To display the Frequency map, click the Calculate frequency button, which can be found on the Statistics toolbar of the Output Window. Similarly, the Density map is created with the Calculate density button. To clear the maps, click the clear button.

Recall that in this model the first half hour is not representative due to a lack of initial boarders. To configure the time interval to be analyzed, click the Output Settings icon. Set the start time to 30 minutes. PD always works in seconds and meters. To set it to 30 minutes, you should either type Mins(30) or express the time in seconds, that is, 30 x 60 = 1800 seconds. Leave the end time at 0, this will select the whole run.

You can also set the sample interval of the Density map (and area, see below). With this parameter you can change the granularity of the density measurements. As discussed above, if you make this time interval short (fine grained), there will be many measurements of short time intervals. This allows you to detect short density peaks that would be averaged out if you would use a longer time interval. On the other hand, because the measurements are based on fewer measurements they are less accurate. If you choose a longer time interval, the measurements will be more accurate but less detailed. In this case, the time interval of 30 seconds is fine.

Beside the granularity in time, you can also change the granularity in space. This can be done by changing the grid size on the Density norms tab of the General Settings window. The General settings window can be opened from the Modeling tab of the Main Menu. The same trade-offs apply as with the time granularity. Here you can also set the colors used to indicate the densities.

The colors for the Frequency map are slightly more complicated because the number of pedestrians that passed depends on how long the model has run. Therefore, the Frequency map usually scales the color thresholds to compensate for that. In the window that appears when you click the Frequency map button, you can configure this: the rescaling can be made linear, logarithmic or it can be disabled, see Figure 5.

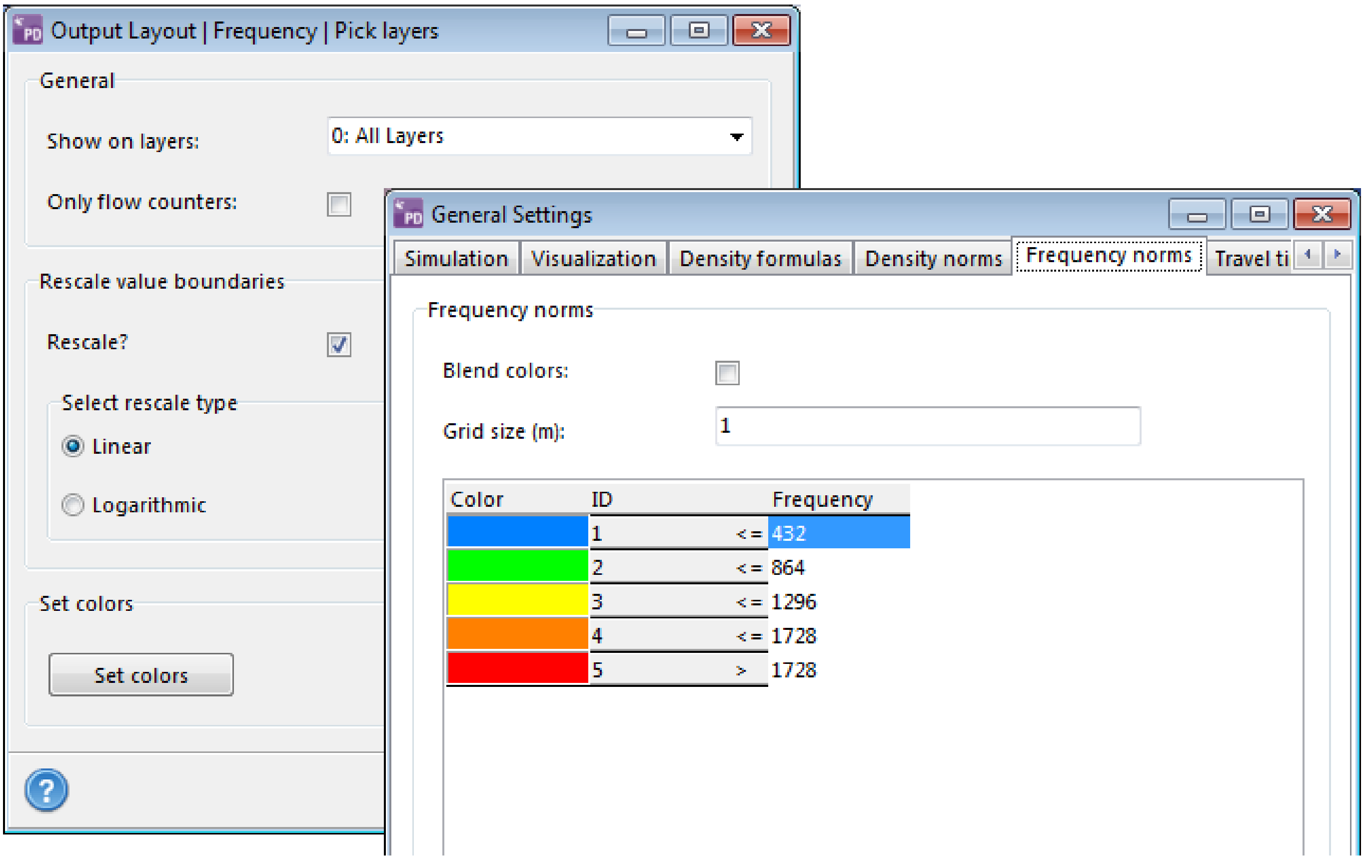


Figure 5: The frequency map dialog. If Rescale is enabled, colors are chosen automatically, either according to a linear or logarithmic scale. If it is disabled, the colors can be configured manually.

5.5 Flow Counters

Flow counters are a convenient tool for measuring traffic streams. A flow counter is a line segment that you draw in the 2D Output window. It indicates the number of pedestrians who crossed that line. For example, the flow counter in Figure 6 is drawn East-West and separately indicates the number of people who crossed it North-South and South-North.

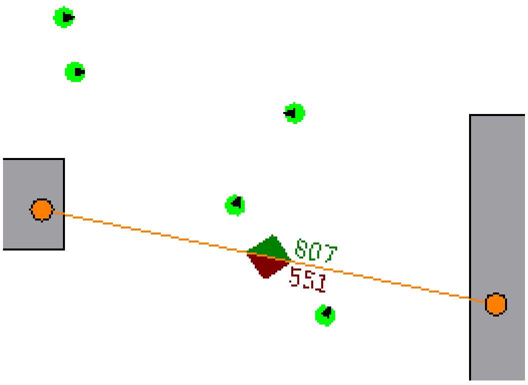


Figure 6: A flow counter showing the number of persons passing either way across the line.

To draw a flow counter, open the *Draw* menu in the 2D Output Window and click the Flow Counter icon. Then select the *line* shape and draw the line north-west across the corridor in the station. You immediately see the counts appear. If you switch to *Select* mode and move the endpoints of the line segment, the counts are updated immediately.

The flow counters have a very nice feature: it is very easy to produce a graph that displays the flow rate across the line segment over time. To do so, switch to *Select* mode and double click the flow counter. The graph will appear immediately, see Figure 7. The graph shows two time series: one for left-to-right, one for right-to-left, where left and right are taken relative to the direction from control point 1 to control point 2. Normally it's easiest to compare the color of the line with the color of the arrows on the flow counter in the 2D Output window.

You can also conveniently zoom in by dragging the mouse inside the graph: dragging upper-left to lower-right zooms in on the selected part, dragging lower-right to upper-left zooms cancels the zoom.

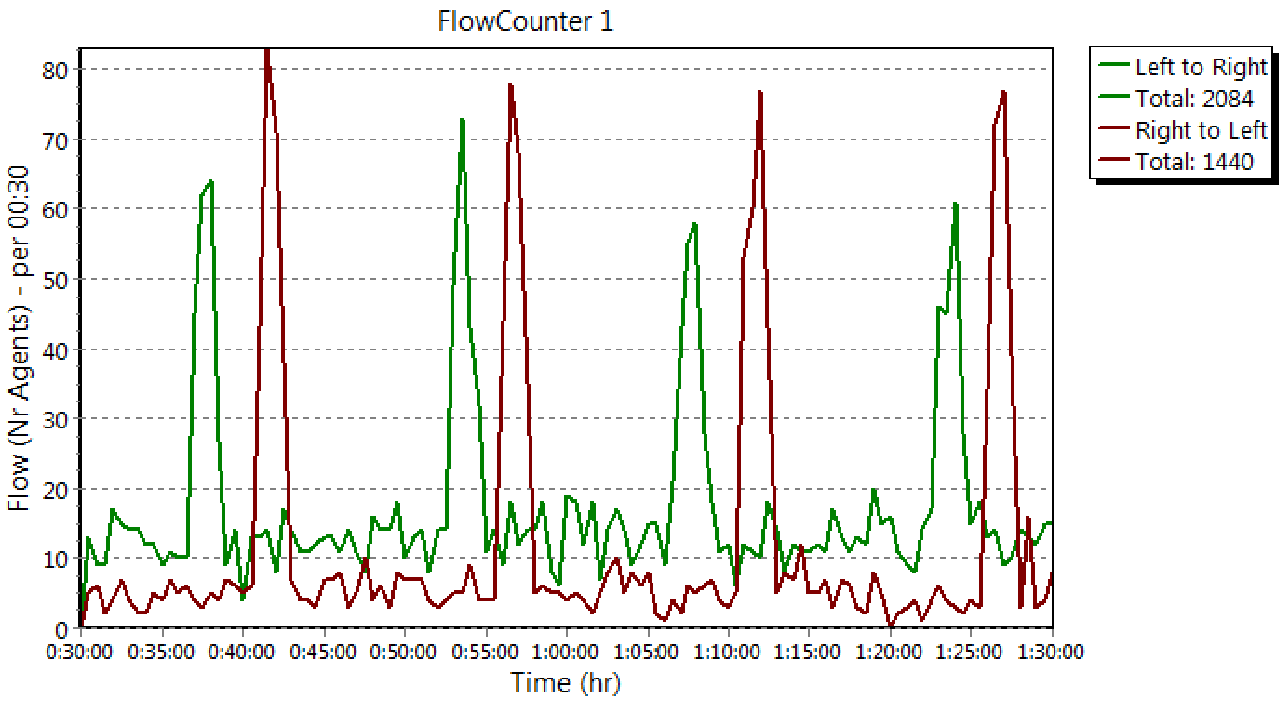


Figure 7: Flow counter graph obtained by clicking a flow counter line. The horizontal axis shows time. The vertical axis displays the flow in Agents per time unit, where the time unit, typically 30 seconds, is configured in the Output Settings.

Exercise 5

Validate the model. Measure the flow in the corridor using flow counters and compare the results with the calculations done in Exercise 1.

5.6 Area Density Histogram

Similar to the flow graph that can be obtained from a flow counter, it is also possible to look at how the density in a specific area evolves over time. Make sure you are in the 2D Output window, click the Area Density icon in the Draw toolbar and choose a shape. Then draw the area in an interesting location in the model. Switch to Select mode and double click the area, a graph similar to Figure 8 will appear.

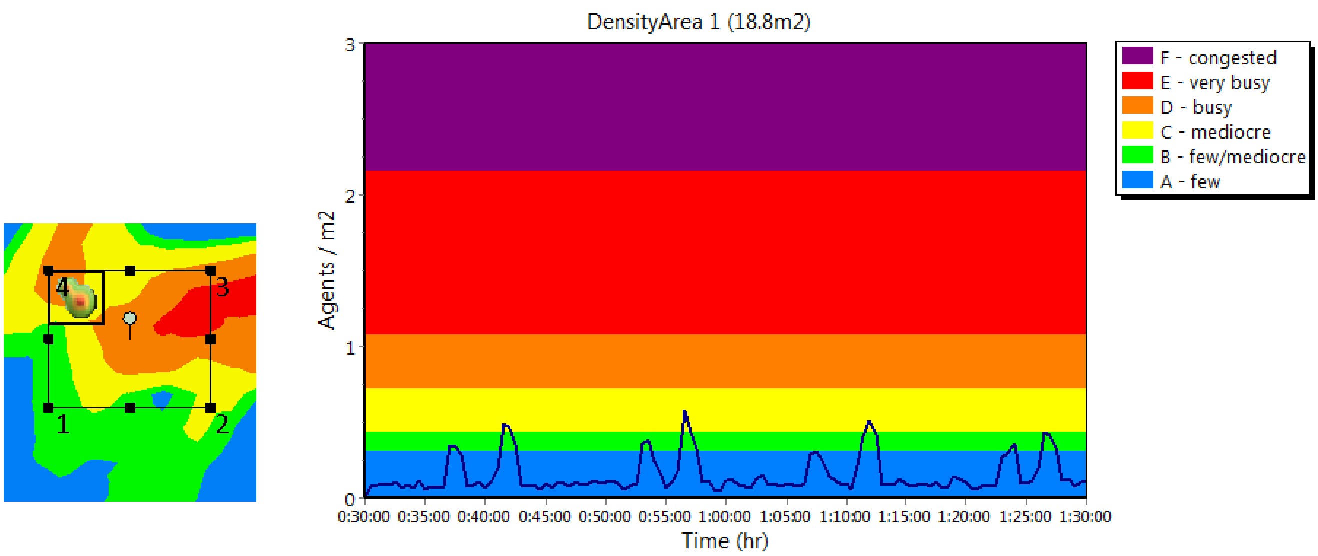


Figure 8: Area Density graph. The horizontal axis shows time. The vertical axis shows the crowd density in Persons per square meter.

The Area Density graph is very useful to investigate whether a purple in the Density map is a case of serious overcrowding or that it is a spike that lasted only a few seconds.

Exercise 6

Examine the dense areas in the station hall and determine which purple areas are seriously crowded and which were spikes. Switch to a longer time interval and explain the effect on the graph.

5.7 Walking times

It is often very important to measure how long it takes for the pedestrians to walk from A to B. For example, in the train station you need to know how long it takes for passengers to transfer from one platform to another during rush hour. Also, in evacuation situations it is important to know which escape routes take longest. Pedestrian Dynamics provides a tool to measure this. In the Statistics toolbar of the 2D Output window, click the Walking Times icon. A window opens in which you can set up one or more routes. A route goes from one activity to another and the graph will be restricted to pedestrians visiting precisely this route. By default, one route is created that goes from any Entry activity directly to any Exit activity. In the train station model this matches all pedestrians.

Select the default route and click Show Graph. A walking time histogram is shown with two very fat bars. This is because the walking times in this model are very short. Close the graph and Edit the default route. Set the Interval (s) to 10 seconds instead of 30. If you save this and view the graph again, the result should be similar to Figure 9.

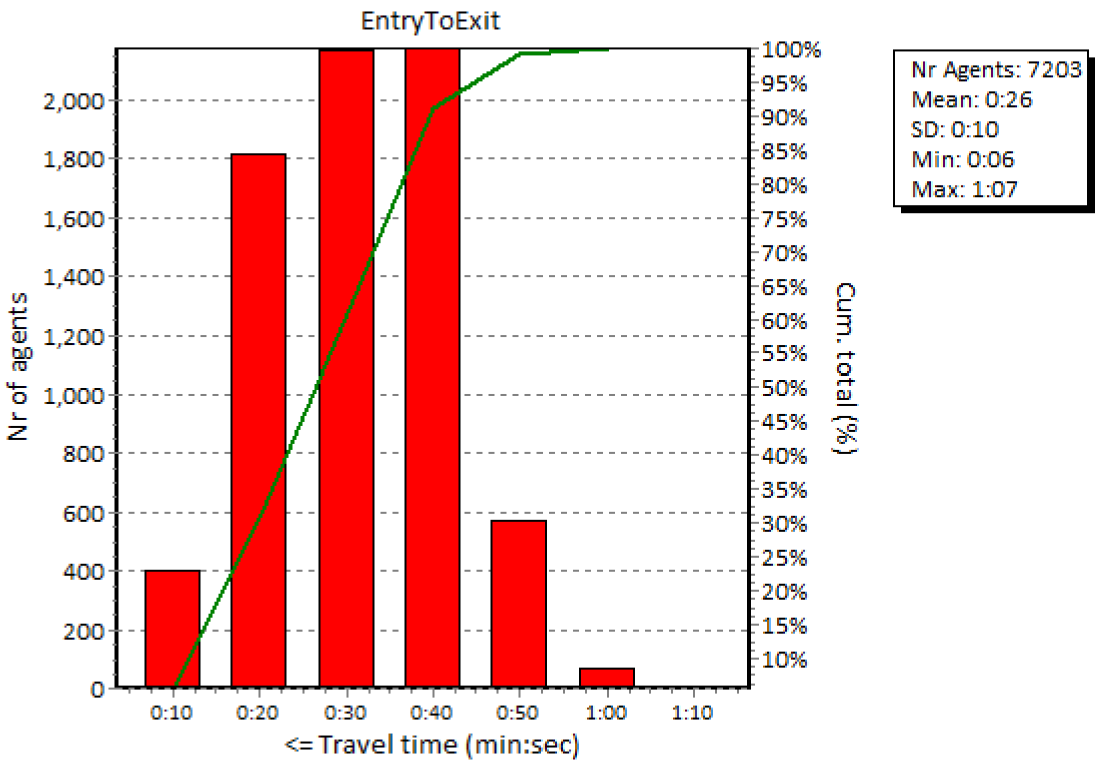


Figure 9: Walking time histogram showing all pedestrian movements between any Entry activity (either Entrance or Platform) to any Exit activity (also either Entrance or Platform). The bar at indicates the agents that took <10s. The second bar indicates the agents that took 10 to 20s, etc.

It is of course more interesting to focus on a specific category of passengers, for example passengers that transfer between platforms.

Exercise 7

Create a walking time histogram for passengers that walk from one platform to another.

Click the Walking times icon in the Statistics toolbar. Click Add to add a new route. Adjust the time interval as with the previous graph. Like the default route, the new route goes from any Entry to any Exit activity, but we can be more specific. Click the Entry activity and in the settings above it, select group Platform. In the dropdown below it, an even more specific selection appears automatically, selecting only Platform 1-North. Adjust this dropdown to select *** *All of this Type & Group* ***. Then click Update. You will see the table row change. Then make the same change in the Exit activity. Click Apply and view the graph.

5.8 Reporting

When comparing the results from multiple replications it is convenient generate the same set of tables, graphs and figures from every replication. You can do this with the Report feature. This tool allows you to specify the information you require, then generate a Quick Report (QRP) of the replication with a single click. You can then easily compare the reports of the replications. By default, the report includes the complete Agent Statistics Table and all Flow Counters, Density Areas and Walking Time Histograms. You can also add screenshots, for example of the Density- and Frequency maps. Click the Report button in the Results tab of the Main Menu to access the Report function.

5.9 Deciding on norms

Before you can conclude whether or not the current plans for the construction work are acceptable, you have to decide on acceptable norms. In general densities higher than 4 persons per square meter in a moving crowd are very dangerous. Even values between 2 and 4 are considered undesirable. However, if you look for example at the ingress process of a sports stadium, higher densities are not uncommon. This can only be safe if high densities only occur in a limited time period and proper crowd management measures are in place, such as stewards who monitor and direct the flow of the crowd.

In a station hall where a lot of pedestrian flows have to cross, densities should be lower to accommodate this. In this case study we will require the density to remain in general below 1 person per square meter, only allowing occasional deviations above this norm. The density is allowed to exceed 1 ped/m2 only 3 times an hour and never longer than 60 seconds. Peaks above 2.5 ped/m2 are not allowed in any case. We base these norms on a measurement interval of 30 seconds and a grid size of 1 x 1 meter.

Exercise 8

Examine the four replications and check whether the norms are met. Describe how you used the tools to arrive at your conclusion.

In the Output settings you can set an Approval Threshold and a maximum Threshold length which will be drawn in the Area Density graphs. The graph will also give an indication of the number of data points that exceed the threshold. However, even with these tools, it is sometimes useful to visually examine a peak in the graph. Depending on the measurements, a density peak with three data points over the critical threshold may look narrow, or a peak with one or two points may look wide. Drag a tall but very narrow box around the peak to zoom in on it. Look for the places where the graph line bends, these are the actual measurements. You can use this to make good judgment of the severity of the peak.

6 Meso vs micro simulation

Pedestrian Dynamics is designed to simulate and evaluate large-scale infrastructures with large numbers of occupants moving through it. Modeling on the so-called mesoscopic scale Pedestrian Dynamics can handle models with crowds consisting of hundreds of thousands of agents. There are however situations in which this mesoscopic approach does not give satisfactory predictions of the pedestrian flow. In these cases the simulation can also be run on the so-called microscopic scale. This approach uses a more detailed representation of the pedestrian movements and provides more accurate results at the cost of higher computational effort.

In this chapter you can find information about different modeling approaches that can be applied in Pedestrian Dynamics such as meso and micro simulation and examples in which microscopic simulation is preferable to mesoscopic simulation.

6.1 Simulation methods for modeling pedestrian flows; macro, meso and micro simulation

In the literature, models for simulating pedestrian flows are typically classified into macroscopic and microscopic models. Macroscopic simulation models focus on the behavior of the crowd as a whole. For example, hydraulic models represent the movement of the crowd as the density and direction of movement in every location in the model, as with a fluid. In microscopic models each pedestrian is represented individually, with its own behavior and taking interactions with other agents into account. Collision avoidance is an important aspect of microscopic simulation models.

A third scale of modeling can be distinguished, the mesoscopic scale. Here the individuality of a pedestrian is maintained but the computationally intensive collision avoidance is omitted. Simulation of pedestrian flows on the mesoscopic scale is therefore suitable for evaluating large-scale infrastructures with many simultaneously moving pedestrians. In many cases this scale of detail provides sufficient information and an acceptable scale of accuracy though in general a mesoscopic simulation model will yield relatively optimistic results compared to a microscopic model.

There are situations in which microscopic modeling is required or preferable, for instance if the specific layout of the environment and the pedestrian movement within it leads to large differences in the results. In these cases the less detailed mesoscopic simulation fails to capture the essential properties of the situation and microscopic simulation gives more accurate results. For example, situations with opposing or crossing flows or situations where the flow is near the full capacity of a bottleneck. Also, when a high level of accuracy is required, micro simulation is preferable.

Pedestrian Dynamics offers both micro- and mesoscopic scales of modeling. You can simply build one model and run it on either the micro- or the mesoscopic scale. You can set this with the *Avoid agent collision* checkbox that can be found on the General settings of the simulate page of the main menu. If this option is checked the simulation will run at the microscopic scale in those parts of the model where the crowd density reaches a certain threshold. If the property *Density threshold* of the General settings is set to zero, every interaction will be modeled at the microscopic scale.

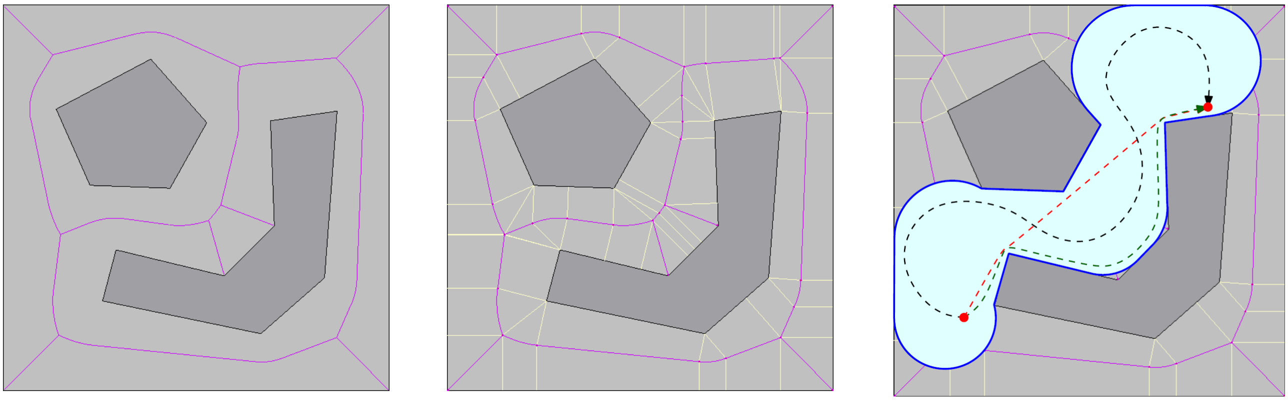
6.2 Modeling human walking behavior

In Pedestrian Dynamics the movement of an agent through the environment is determined by its Activity Route, a list of Activities the agent plans to perform within the environment. For each Activity you can define at which Activity Locations the activity can be performed. The agent will choose one of the Activity Locations based on the *Location distribution* defined on the Activity. The next step is that the agent picks a specific point within the location, based on the *Location Approaching* property of that location. The agent has to reach that point first before the activity can be performed.

The agent has to find a route from its start position to the goal position in the next activity location. To do so, the agent first chooses a global route through the environment. When traversing this route local circumstances will influence the agent and determine the eventual path. This local behavior can be modeled on the microscopic or the mesoscopic scale. In general agents try to walk with their own desired speed which is determined by a probability distribution on the Agent Profile. However, local circumstances such as the terrain and in particular the presence of other agents may force them to move slower or to change direction.

Determining the global route

In Pedestrian Dynamics the Explicit Corridor Map (ECM) is used to find a global route from the start position of the agent to the desired goal position. This map describes the whole walkable space and is automatically created when a simulation run is started. An important part of the map is the *medial axis,* a set of curves describing the middle of the walkable space, see Figure 1a. The medial axis represents the global structure of the environment. When routing, the start and goal position of the agent are first mapped to points on the medial axis. Then the route along the medial axis between these points is computed based on the *least effort* principle. This is not yet a realistic path for a pedestrian, but it does describe a corridor through which the agent can reach its goal, see Figure 1c. Within this corridor there are many possible paths from the start to the goal position. For instance, the agent can choose to stay on the left or right side of the corridor, or to follow the shortest possible path. The so-called *indicative route method* (IRM) selects a realistic path through the corridor, the indicative route. The indicative route is the global route the agent will try to follow to reach its goal.



(a) Medial axis (b) Explicit Corridor Map (c) Corridor

Figure 1: (a) The medial axis, shown in purple, runs through the middle of the walkable space. (b) Closest-point annotations, shown in yellow, turn the medial axis into the ECM navigation mesh. (c) The corridor, shown in blue. The dotted lines show several possible routes through the corridor from the start position to the goal.

The precise cost function that is used in the algorithm to compute the least effort route along the medial axis can be found in the documentation. This cost function takes into account the crowdedness along the route and the environment. For example, a flight of stairs is less attractive to use compared to an escalator and an agent will walk slower on a tilted area. These environmental influences are usually expressed as a *discomfort factor*. This is a property of stairs, escalators and other infrastructural elements and can be altered by the user.

The parameters that are used to find the global route along the medial axis and to find an indicative route can be found on the first three pages of the *Agent profile*. The agent profiles can be created and edited from the Agent Input Settings that can be opened from the Simulate page of the main menu. For example the *viewing distance* is the distance along which the agent can take crowded areas into account. You can also enable the *re-route periodically* option and set a time period such that the agents will reconsider their global route at the start of each new period. An agent might switch to a different route if the crowded areas have changed or if the agent has become aware of a crowded area that is now within its *viewing distance*. On the *route following* page of the Agent profile, parameters that are used to find an indicative route can be found. The *side preference* gives a bias towards one side of the corridor, while the *preferred clearance* gives the distance the agent tries to keep from the obstacles.

At the start of each simulation run the Explicit Corridor Map (ECM) that is used to determine global routes through the environment is computed. You can also create and clear this map manually using the buttons on the Simulate page of the main menu.

On the Display page of the main menu in the section *Show ECM Network* you can find buttons that allow you to toggle the visibility of different elements of the ECM navigation mesh, as depicted in *Figure 1*. From left to right you find a button for the medial axis, vertices, the nodes and the closest points, see Figure 1b.

If you have an extremely large model it might take a few minutes to create the ECM network. Once a network has been created it can optionally be saved to a file. If saved under the same name as the model it will be loaded automatically. If you then run a simulation the ECM is not recomputed but the loaded network will be used. You can save the network using the buttons in the *ECM Network* section of the File page of the main menu.

Local behavior in micro models

There are many different ways to model the local behavior of agents. The local behavior models the effect of interaction with other agents. Pedestrian Dynamics uses a *vision based* approach. Each agent has a *field of view* that determines the part of the environment that the agent can perceive directly. It is a cone with the origin at the agents position. Only agents that are within the field of view will be taken into account for the collision avoidance. While the agent moves along the indicative route towards the final goal at each simulation step small adjustments in speed and direction are made to avoid these collisions. The algorithm used to determine the adjustments takes the distance to the first collision and the deviation from the direction towards the local goal into account. The algorithm selects the speed and direction to find the most energy efficient way to avoid a collision. Changes in velocity and speed require energy, making minor detours sometimes more efficient than the shortest path.

The parameters used for the local walking behavior can also be defined on the Agent profile. Most of them determine the *Field of View* (FoV).

Local behavior in meso models

On the mesoscopic scale the speed of the agents is also adjusted each simulation step, again to model the effect of interaction with other agents. These adjustments are made on a much coarser scale than on the microscopic scale. The speed is adjusted not because of interaction with specific nearby agents but only based on the general local crowdedness. While the agent is following its global indicative route, at each simulation step a new speed is determined based on a speed-density relation. The ECM keeps track of the current number of pedestrians in each of the polygonal areas in the model. In Figure 1b these areas are marked with white lines. These local density counts are used to determine the local density around an agent.

Because agents do not avoid specific other individuals but only react to the local crowdedness, it can happen that the agents seem to move through each other. This may seem unrealistic but the effect of agents avoiding each other is in fact incorporated in the speed-density relation that governs their walking speed, giving a good first estimate of this effect. Although running the simulation on a mesoscopic scale is less realistic than running it on a microscopic scale the results obtained can still be useful. Especially if you are mostly interested in measuring the throughput or the density build up the mesoscopic scale can be sufficient.

References

The techniques described here to model the walking behavior are based on scientific research and can be found in literature. PD uses efficient crowd simulation algorithms and software, developed together with Utrecht University (UU) in Utrecht, The Netherlands [1]. The interested reader can find more information about the Explicit Corridor map (ECM) in [2], [3] and [5]. For more information on the indicative route method see [4]. The collision avoidance algorithm that PD uses is based on the vision based model developed by Moussaïd, Helbing and Theraulaz [6].

[1] Utrecht University - http://www.uu.nl/EN/

[2] R. Geraerts. “Planning Short Paths with Clearance using Explicit Corridors.” In *IEEE International Conference on Robotics and Automation (ICRA'10)*, pp. 1997-2004, 2010.

[3] W.G. van Toll, A.F. Cook IV, and R. Geraerts. “Navigation Meshes for Realistic Multi-Layered Environments.” In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'11)*, pp. 3526-3532, 2011.

[4] I. Karamouzas, R. Geraerts, and M. Overmars. “Indicative Routes for Path Planning and Crowd Simulation”. In *The Fourth International Conference on the Foundations of Digital Games (FDG'09)*, pp. 113-120, 2009.

[5] W.G. van Toll, A.F. Cook IV, and R. Geraerts. “Real-Time Density-Based Crowd Simulation.” *Computer Animation and Virtual Worlds (CAVW)*, 23(1):59-69, 2012.

[6] M. Moussaïd, D. Helbing, G. Theraulaz. "How simple rules determine pedestrian behavior and crowd disasters." *Proceedings of the National Academy of Science(PNAS)*, 2011.

6.3 Comparing micro and mesoscopic level simulation

As we described in the previous sections, the mesoscopic scale describes the local behavior of the agents on a coarser level than the microscopic scale. Although the mesoscopic scale is less detailed, a simulation run on this scale still maintains a good approximation of local pedestrian behavior. Since at the mesoscopic scale the computational cost is much lower, the mesoscopic scale is suitable for evaluating large-scale infrastructures with many simultaneously moving pedestrians. There are however situations in which the mesoscopic scale does not give realistic predictions. The largest differences can be expected where there are a lot of interactions between agents, for instance when there are opposing pedestrian flows or if the flow is near the capacity of a bottleneck. If the goal is to measure the throughput or the density build up, simulation at the mesoscopic scale will typically be sufficient but if the goal is for example to measure the average travel time, the inter-agent interactions must be taken into account. In this section we will examine in several such situations the difference in results obtained at the meso- and microscopic scale.

Pedestrian Dynamics has the capability of selectively enabling micro simulation once the density exceeds a certain threshold. This can be controlled using the *Avoid agent collisions* checkbox on the General Settings page. In all experiments described here, micro simulation is either completely turned off or completely turned on (with threshold=0).

Counterflow

It makes intuitive sense that more agent interaction occurs when two or more pedestrian flows overlap that do not share the same direction. Of course this also depends on the flow, in a sparse flow there are fewer interactions than in a dense flow. The more interactions, the larger the deviation we expect to see between the simulation results at the two scales. We will now describe some experiments we have run to examine these differences. In the first experiment we look at a situation with unidirectional flow at increasing densities, expecting the difference in travel times between micro and meso to increase with the density. In the second experiment we look at counterflow, two pedestrian flows in opposite directions in the same corridor. In this experiment we keep the number of pedestrians constant and vary the ratio between the two flows.

Travel time is the primary performance metric considered in these experiments. We will examine the micro/meso sensitivity, that is, the ratio between the average travel times of a series of micro scale simulations and a series of meso scale simulations.

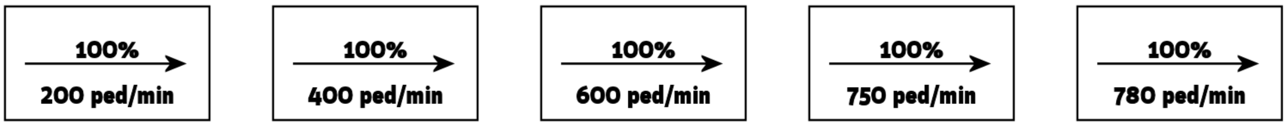


Figure 2: Overview of parameter settings for experiment 1 with unidirectional flow. For each experiment we increase the in-flow.

In the first two experiments we consider a corridor that is 200 meters long and 10 meters wide. The average walking speed of the pedestrians is set to 1.35 m/s and the effective walking distance in the corridor is 192 meter leading to an expected travel time of about 142 seconds if there is no interaction between the agents at all. In each experiment we measure the average travel time from one end of the hall to the other. We ignore the first two minutes of the run, during which the corridor is still filling up. Each experiment is performed on the mesoscopic scale as well as the microscopic scale. For each scenario we measure the average travel time to compute the micro/meso sensitivity.

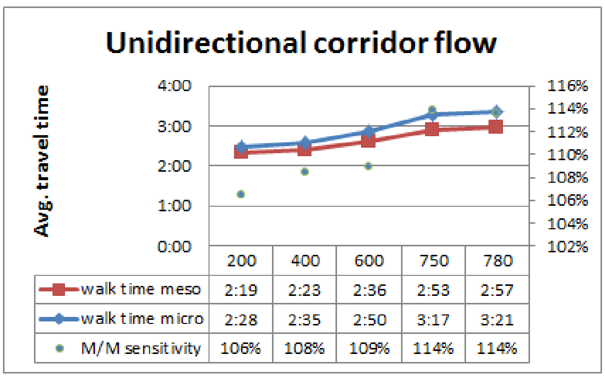


Figure 3: Results of Experiment 1: unidirectional flow

In the first experiment we consider the situation in which all pedestrians walk in the same direction. From the literature we know that in a 10m wide corridor the maximum flow should be somewhere around 800 ped/min. We run the experiment varying the rate of pedestrians we introduce in the model, see Figure 2 for an overview of the different in flows, and Figure 3 for the results. At 200 ped/min we measure an average walking time of 139 seconds in the meso simulation and 148 seconds in the micro simulation which gives a deviation of about 6%. At the highest of 780 ped/min the deviation increases to 14%.

You might not have expected such large differences even with unidirectional flow. The reason is that even with *unidirectional flow the pedestrians walk at different preferred speeds. This means that in denser flows we get a lot of agent interactions because faster pedestrians overtake slower ones.*

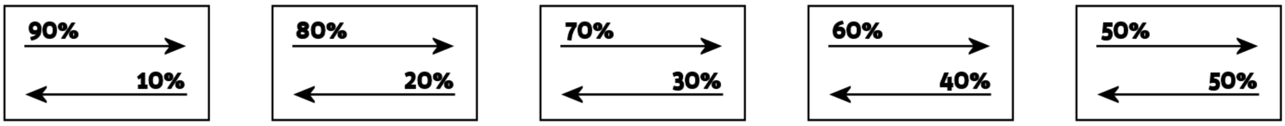


Figure 4: Overview of parameter settings for experiment 2 with bidirectional flow. We keep the total flow fixed at 400 ped/min and change the ratio between the opposite flows.

In the second experiment we introduce counterflow. We keep the total flow fixed at 400 ped/min and vary the number of pedestrians walking in the opposite direction, see Figure 4. Note that the West-East flow is always denser than the East-West flow. With 100% of the pedestrians walking West to East this reduces to the previous experiment, at lower percentages we have counterflow. We would expect the effect of enabling micro simulation to be much larger for the sparse flow than for the dense flow and that turns out to be case. With the dense flow at 90% and the counterflow at 10%, the difference between meso and micro is 15% for the dense flow and a staggering 42% for the sparse flow. When the dense flow reduces to 60% (40% counterflow) the percentages become 25% and 31%, respectively. This makes sense because the counterflow gives rise to a lot of interactions but the two flows are now more alike.

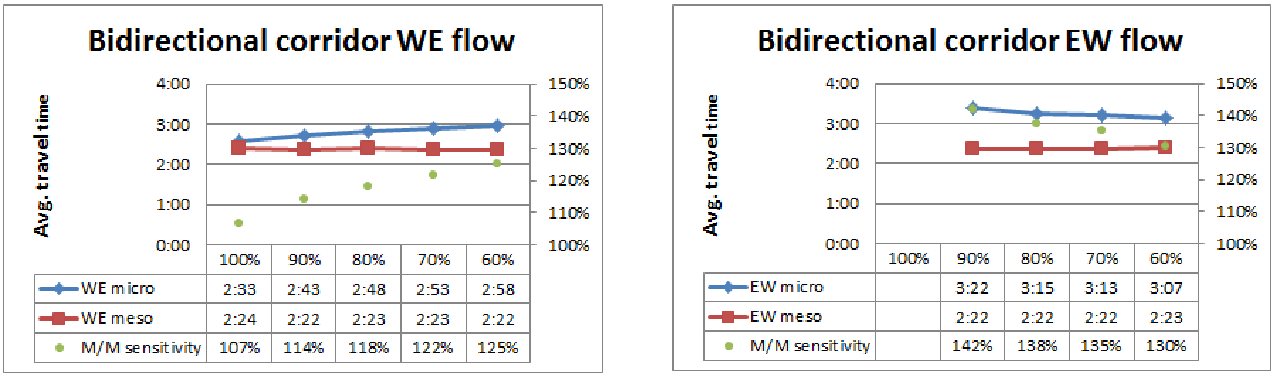


Figure 5: Results of Experiment 2: bidirectional flow.

Flow at cross sections

We examine two types of crossing traffic. In experiment 3 we look at two unidirectional flows that cross perpendicularly. We vary the relative proportions of traffic traveling West-East (WE) and North-South (NS), see Figure 6. In experiment 4 we have a dense bidirectional stream flowing around the corner (50% WS and 50% SW) and a sparser bidirectional stream 50% WE and 50% EW. Again we vary the proportions of the streams, see Figure 7.

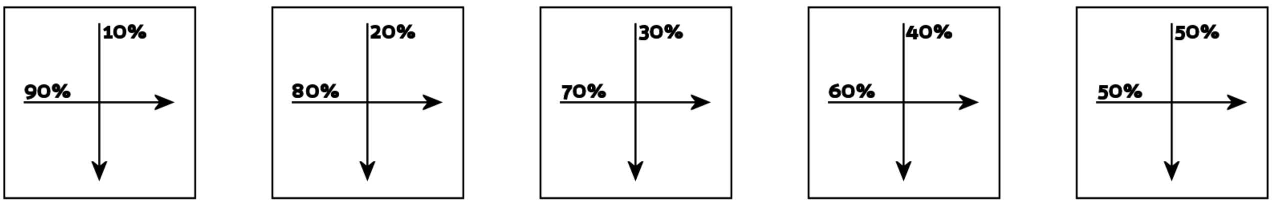


Figure 6: Overview of parameter settings for experiment 3 with two perpendicular unidirectional flows. Keeping the total flow fixed at 200 ped/min we vary the ratio between the WE and NS flow.

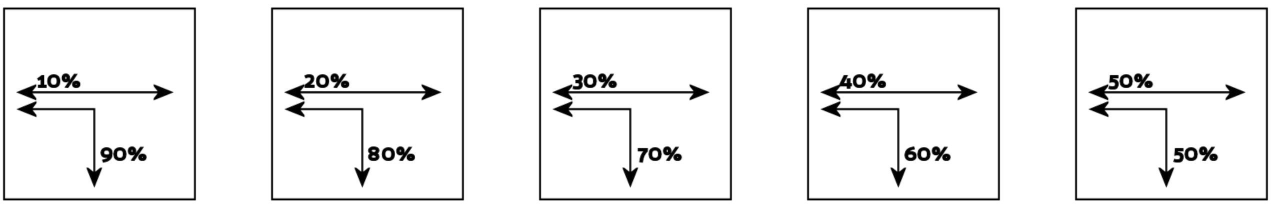
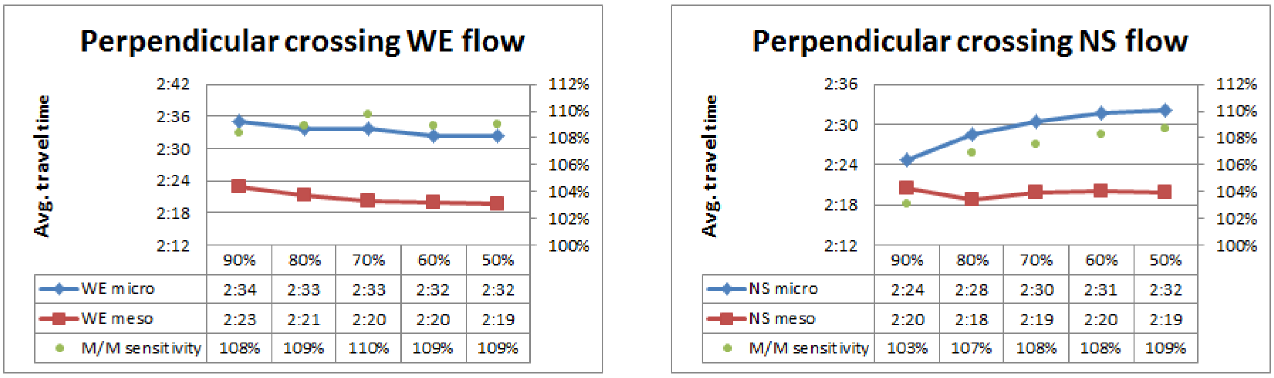


Figure 7: Overview of parameter settings for experiment 4 with a WS bidirectional corner flow and a WE bidirectional straight flow. Keeping the total flow fixed at 200 ped/min we vary the ratio between the two flows.

For the WE traffic in experiment 3 the micro/meso sensitivity remains stable at around 9%, see Figure 8. This despite the fact that traffic in the WE direction drops from 360 to 200 ped/min. For comparison, in the unidirectional corridor experiment we saw 8% at 400 ped/min and 6% at 200. This indicates that the more frequent encounters in the 10x10 square meter crossing area compensate for the reduced crowdedness in the rest of the model.

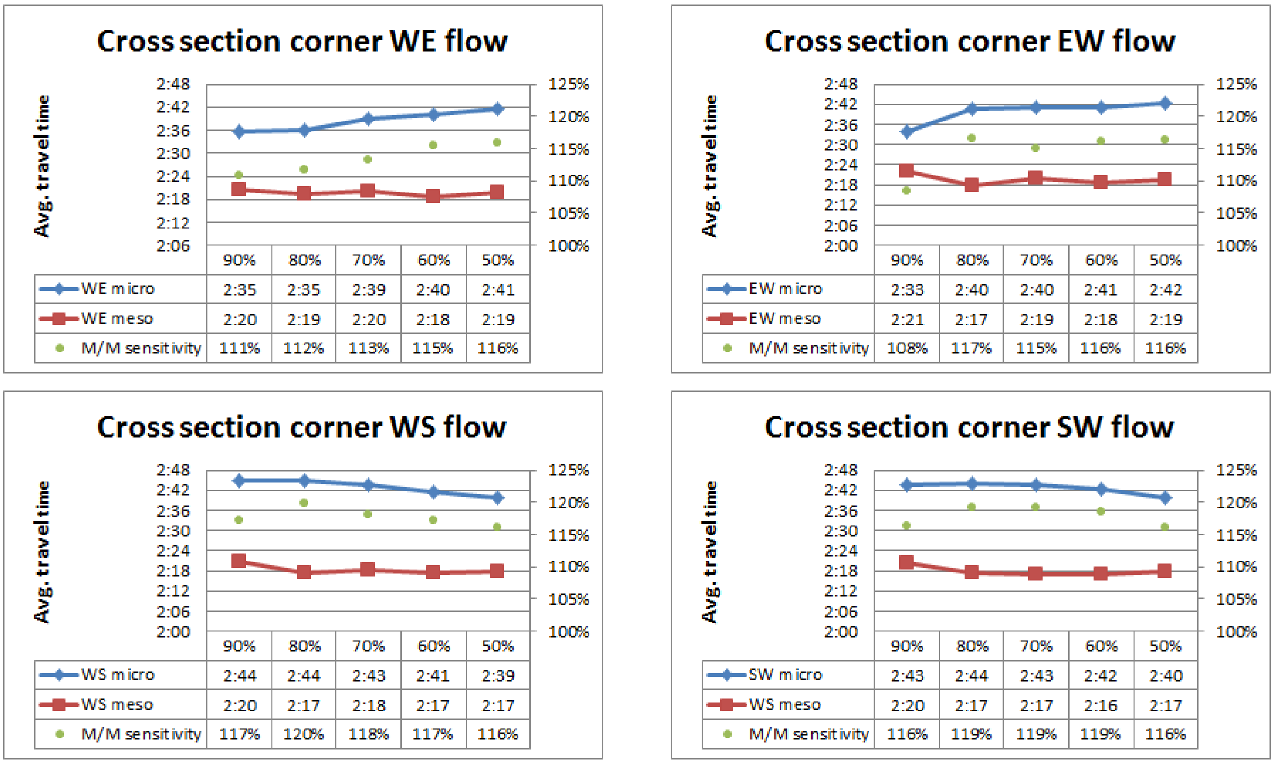


	90%	80%	70%	60%	50%
WE micro	2:34	2:33	2:33	2:32	2:32
WE meso	2:23	2:21	2:20	2:20	2:19
M/M sensitivity	108%	109%	110%	109%	109%

	90%	80%	70%	60%	50%
NS micro	2:24	2:28	2:30	2:31	2:32
NS meso	2:20	2:18	2:19	2:20	2:19
M/M sensitivity	103%	107%	108%	108%	109%

Figure 8: Results for Experiment 3, two perpendicular crossing flows.

In experiment 4 we see an increase in the micro/meso sensitivity of the sparse WE traffic, see Figure 9. When the ratio between the WE traffic and the WS traffic is 10:90, i.e., the WS traffic is much denser than the WE traffic, the average walking time measured on a microscopic scale is about 11% higher while at a ratio of 50:50 where the two streams are equally dense we get about 16%. It is not surprising that we get an increase in the deviation at the 50:50 ratio because at that ratio the EW traffic has counterflow in the whole corridor at this ratio while if it is sparse there is only counterflow where it overlaps with the other flow. For the WS traffic the maximum sensitivity occurs at 80:20.



	90%	80%	70%	60%	50%
WE micro	2:35	2:35	2:39	2:40	2:41
WE meso	2:20	2:19	2:20	2:18	2:19
M/M sensitivity	111%	112%	113%	115%	116%

	90%	80%	70%	60%	50%
EW micro	2:33	2:40	2:40	2:41	2:42
EW meso	2:21	2:17	2:19	2:18	2:19
M/M sensitivity	108%	117%	115%	116%	116%

	90%	80%	70%	60%	50%
WS micro	2:44	2:44	2:43	2:41	2:39
WS meso	2:20	2:17	2:18	2:17	2:17
M/M sensitivity	117%	120%	118%	117%	116%

	90%	80%	70%	60%	50%
SW micro	2:43	2:44	2:43	2:42	2:40
WS meso	2:20	2:17	2:17	2:16	2:17
M/M sensitivity	116%	119%	119%	119%	116%

Figure 9: Results for Experiment 4, a WS bidirectional corner flow and a WE bidirectional straight flow.

Bottleneck situations

Here we examine a bottleneck situation. We will see that the differences between simulation at the micro and meso scale are especially large when the bottleneck nears its maximum capacity. In fact, only micro scale simulation is able to predict the capacity. In this experiment, pedestrians move through a 20 meter long corridor with a width of two meters. They approach the bottleneck from an open area, see Figure 10. We have placed several flow counters in the corridor to measure the average flow. The simulation is run for 10 minutes of which the first two are ignored in the measurements of the average flow because the flow has not yet stabilized, see

Figure 11.

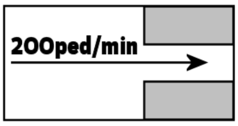
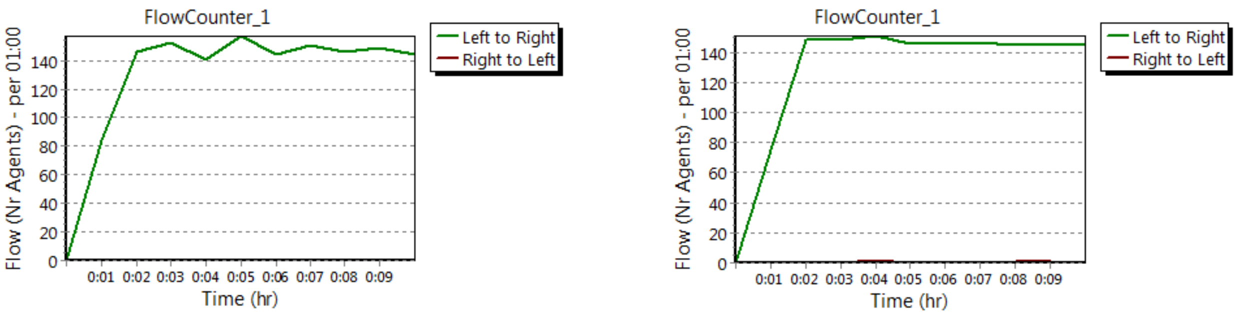


Figure 10: Situation sketch of Experiment 5, flow at a bottleneck.

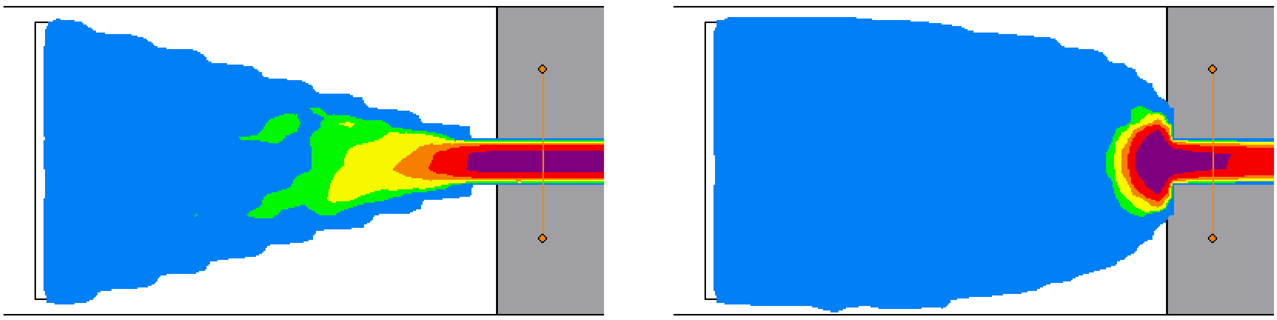


(a) Flow at meso scale

(b) Flow at micro scale

Figure 11: The simulation takes about 2 minutes to reach a steady flow rate.

The difference between micro and meso scale is clearly visible in the density maps in Figure 12. We see that the micro level simulation in Figure 12b correctly predicts the pedestrians clogging near the entrance of the corridor and pedestrians trying to approach it from the sides whereas the meso picture shows how the pedestrians essentially follow a straight path from wherever they began to the entrance and only adjust their walking speed, not their path. As a consequence, in the meso simulation the highest densities occur inside the corridor instead of before it.

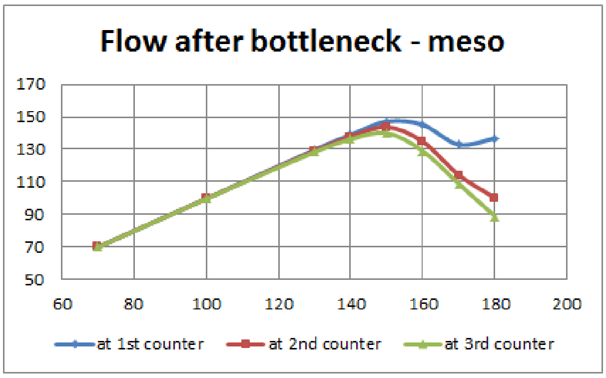


(a) Density map at meso scale

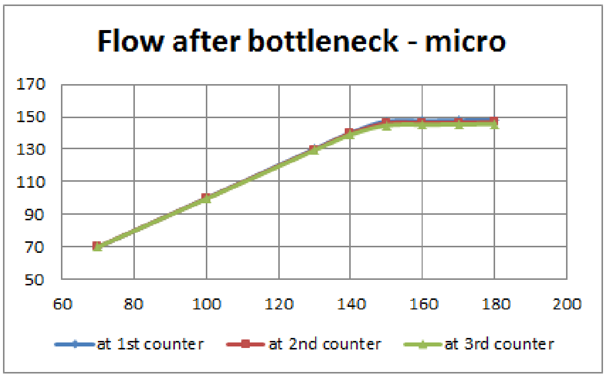
(b) Density map at micro scale

Figure 12: Density map of the bottleneck at an arrival rate of 150 ped/min, (a) at meso scale, (b) at micro scale.

It is also interesting to compare the flow rates further inside the corridor, as you can see in Figure 13. In the micro simulation clogging occurs before the corridor and the flow rate remains the same all through the corridor because the pedestrians can walk at a reasonable pace once they have passed the entrance. In the meso simulation the high density leads to slower walking speeds which in turn lead to even higher density. This can be seen clearly in Figure 13a where the flow at the first flow counter is significantly higher than at the third because the walking speed in the corridor has become so slow that many pedestrians have not yet been able to reach the third flow counter.



(a) Flow results at the meso scale



(b) Flow results at the micro scale

Figure 13: Pedestrian flow at three positions along the corridor, (a) at meso scale, (b) at micro scale.

If we look at the results for the third flow counter we see that for the microsimulation run the flow increases with the number of pedestrians until an average capacity of around 70 ped/min is reached. In the mesoscopic runs the flow increases until around 65 ped/min. Due to the crowded area before and in the beginning of the bottleneck the flow there is already decreased allowing lesser agents to move past the third flow counter.

6.4 Conclusion

In this chapter we have learned that PD offers two scales of modelling, the mesoscopic scale and the microscopic scale. With the General settings *Avoid agent collision* checkbox one can easily switch between the two. The mesoscopic scale describes the local behavior of the agents on a coarser level than the microscopic scale. Although the mesoscopic scale is less detailed, a simulation run on this scale still maintains a good approximation of local pedestrian behavior. Since at the mesoscopic scale the computational cost is much lower, the mesoscopic scale is suitable for evaluating large-scale infrastructures with many simultaneously moving pedestrians.

There are situations in which the mesoscopic scale does not give realistic predictions. We have seen that the largest differences occur where there are a lot of interactions between agents, for instance when there are opposing pedestrian flows or if the flow is near the capacity of a bottleneck. In these cases the micro/meso sensitivity, that is, the ratio between the average travel times of a series of micro scale simulations and a series of meso scale simulations, can rise up to 142%.

When modelling its always a good approach to keep it simple and only take more detail into account when required. In many cases the mesoscopic scale will be sufficient and otherwise one can add more detail switching to micro scale by checking the *Avoid agent collision* check box. One always has to investigate if this is sufficient. If so then PD also has the capability of selectively enabling micro simulation once the density exceeds a certain threshold.