TECHNICAL UNIVERSITY DELFT
FACULTY EWI

Modeling passenger flows on a train platform with obstacles

Emmy Thans
June 22, 2017

Abstract

In this report, a model for modeling passenger flows on a platform with obstacles is presented. The report incorporates situations with one exit, benches as obstacles and an additional train and compares those to a basic model in which the platform has two exits and no obstacles. The present study considers the Particle Model as a basis for passenger flows on a platform. The parts of the Particle Model that are integrated are chemotaxis, random walk, repulsive forces and mechanotaxis. Therefore the model is based on both deterministic and stochastic principles. Simulation results are discussed concerning the time needed for a certain amount of passengers to leave the platform, the average number of encounters between individuals and of individuals with benches.

1 INTRODUCTION

Traveling by train is done on a daily basis by over a million people in the Netherlands [1], either for economical or leisure reasons. However, what makes a passenger, once on the platform, move around and in what direction? The goal of this study is to investigate this issue quantitatively by programming and animating passenger flows on a platform as closely resembling reality as possible. Modeling passenger flows can give insight in how and why people move, and more importantly, the efficiency of passenger flows under various circumstances can be quantified. The flows that are created can provide information on what does and does not enable a smooth transition of people from a train to the platform and off it again.

The report starts off by introducing the model that serves as a basis for the whole research, the Particle Model, in Section 2. It is explained what this model entails, which parts of it are relevant to passenger flows and in what way they are applicable. Thereafter the parts of the Particle Model that is used to build the model for the passenger flows will be translated to a mathematical model in Section 3 by making use of numerical methods. Based on those formulas the actual programming can commence. In this case Python was chosen as the programming language, since it is easy to read and understand and it is able to provide good animations. How the mathematical model was translated to a Python code is explained in Section 4. The full Python code is added as an appendix at the end of the report. Since animations can not be shown in a report, Section 5 contains snapshots of the relevant animations as well as the corresponding results. The results consist of a basic model and three subtopics: one exit, benches as obstacles and an extra train. Each of these subtopics is compared to the basic model in which there are two exits and no obstacles. The report ends with a discussion on the process and final conclusions in Sections 6 and 7 respectively.

2 PARTICLE MODEL

The Particle Model is a way of describing the behavior and migration of living cells through various angles. The model can be applied to a numerous amount of situations. As of now the Particle Model has mainly been applied biologically and medically. In this report that application is stretched to something many people encounter every day: human beings on a platform. We finally remark that this class of model may also be applicable to, for example, modeling the evacuation of people out of concert halls under normal or under stressful circumstances, such as during a fire or a terrorist attack.

2.1 PARTICLE MODEL APPLICABLE COMPONENTS

There are many ways to describe the behavior of living cells. A few of these ways are applicable to the behavior human beings portray when on a platform of a train station: chemotaxis, random walk, repulsive forces and mechanotaxis. Before we can begin to discuss why and how these are applicable, a mutual understanding of them must be created first:

- **Chemotaxis**: the migration of cells caused by a concentration gradient [2][3][4]. E.g. cells will move towards an area that has a large(r) concentration of certain chemicals, such as nutrients or oxygen. Cells may also migrate opposite to the gradient in case of toxins.
- **Random walk**: a stochastic process for migration. It almost never happens that a cell moves directly and only in the right direction, i.e. in the direction where it wants to go. A random variable needs to be added to the model to include fluctuations in the movement of cells [3][5].
- **Repulsive forces**: these forces occur whenever there is physical contact between cells or they come within a specific vicinity of each other [3]. They make the cells move away from each other [5].
- **Mechanotaxis**: the migration of cells caused by a mechanical signal [2][6]. E.g. a hormone secreted by an organ in your body can make cells move and/or execute a specific task.

It is clear that the descriptions provided above are all focused on cells and their movement. However, passenger flows on a platform involves human beings instead of cells. Even though one might think that people and cells are completely different things, they are not. In particular, people's behavior on a platform can be well described using the above mentioned components of the Particle Model. How this can be done exactly is explained in the next Subsection.

2.2 APPLYING THE PARTICLE MODEL TO PASSENGER FLOWS ON A PLATFORM

As has been stated before, the Particle Model is very versatile and can be translated to many different situations, including passenger flows on a platform. The way in which this can be done logically, using the four components of the Particle Model mentioned in the previous Subsection, is as follows:

- **Chemotaxis**: in the present model, there is no chemical of which the gradient triggers migration of individuals over the platform. As a passenger alights the train, then (s)he looks for the exit of the platform. In principle the individual will likely choose the shortest path to the exit. This introduces a deterministic term to the migration of the individual. This is in line with chemotaxis, which also involves a deterministic component in the overall migration vector.
- **Random walk**: imagine you have not paid attention and are not sure where you are when you get out of the train or where to go next. You are not likely to go directly into the right direction.

For that reason a random variable, or random walk component, needs to be added to simulate a feeling of 'being slightly lost or unsure'. This mode to migration entails the stochastic variations to migration as a result of sleepiness, haziness or as a result of certain levels of intoxication.

- **Repulsive forces**: apart from couples, people generally do not enjoy walking side-by-side touching each other. For that reason there is a certain 'threshold' or 'comfort zone' that needs to be maintained in order for people to comfortably move around. However, due to crowdedness this is not always possible. Therefore people might slightly deviate from their original path to not bump into other people too often or preferably not at all.

- **Mechanotaxis**: however lost passengers may be when arriving on a platform, there are always things to help them out. Examples are announcements, signs, lights, etc. These can all help to find the right way to the exit, consequently making one move in that direction. Intuitively, it seems logical that the stronger the presence of mechanotactic processes, the smaller the influence of the random walk component. Like in the case of the desire to use the shortest path towards the exit of the platform, this migration mode is deterministic, though it can be switched to a stochastic mode to incorporate possible sudden changes on the platform, as a result of for instance violence, change of time-tables due to damage on trains or train-passenger collisions.

Now that the components that will be used as a basis for the mathematical model have been explained, their numerical implications can be researched. Section 3 shows what these implications exactly are per component.

3 MATHEMATICAL MODEL

For the sake of clarity, less running time and aesthetics, the model was simplified by making the following assumptions:

1. The passengers can move away from each other without necessarily touching. This can be considered as a comfort-zone: every individual likes to be at a certain distance from others.

2. The passengers are able to move freely across the platform (apart from bumping into other passengers or any other type of obstacle).

3. The passengers are projected as 2D-circles on the platform. This simplifies calculations and adds to the clarity of the model. It also concurs with reality in the sense that people only move on a 2D-plain when on a platform: they do not move up or down (apart from when using stairs or jumping up and down, but these are left out of the model used in this study).

4. The size of the passengers is predefined and does not change during animation. Furthermore, all the passengers have been given the same size for the sake of simplicity.

5. Every passenger wants to exit the platform. No one stays behind to wait for another train to arrive at that platform. Furthermore, the passengers want to exit through the nearest exit.

6. At some point every individual has figured out where the nearest exit is and will therefore not falter to walk into the right direction. This will be referred to as the 'green-zone' and will be elaborated on later.

Now that the initial assumptions have been listed, every part of the Particle Model that is used can be further analyzed on a mathematical level.

3.1 CHEMOTAXIS

The exits of the platform can be regarded as sources of attraction to the people: this is where they want to go. This is what mainly determines their initial direction and with that their displacement, also denoted as $\mathbf{u}(\mathbf{x}(t))dt$ in which $\mathbf{x}(t)$ and $\mathbf{u}(\mathbf{x}(t))$, respectively, represent the position and velocity (vector) at time t. That being said, the new position for individual i can be written as

$$d\mathbf{x}_i(t) = \mathbf{u}_i(\mathbf{x}_i(t))dt.$$

in which

$$\mathbf{u}(\mathbf{x}(t)) = \frac{\mathbf{g}-\mathbf{x}(t)}{\|\mathbf{g}-\mathbf{x}(t)\|} \cdot v_{walk},$$

$\mathbf{g}$ stands for the location vector of the goal of each individual, which is an exit of the platform, and v_{walk} is the walking speed. Figure 3.1 provides a visual sketch of the displacement of an individual. Note that the direction in which an individual moves, $\mathbf{g}-\mathbf{x}(t)$, is normalized. This is done to have more control over the actual speed and the displacement of that individual. Otherwise, all passengers would move slower the closer they get to their goal. Normalization does not change the direction and enables the addition of an 'individual' walking speed v_{walk}.

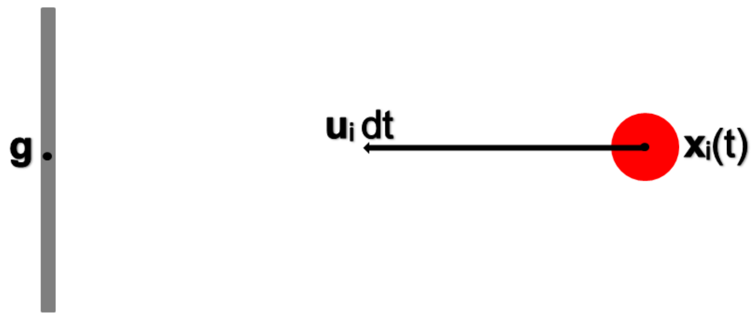

Figure 3.1: Goal $\mathbf{g}$, position $\mathbf{x}_i(t)$ of the red circle, or individual i, at time t and displacement $\mathbf{u}_i$ of individual i.

Unless an individual is familiar with the station (s)he has arrived at and knows where on the platform (s)he will end up when leaving the train, (s)he is not likely to directly walk into the right direction. In other words, a possibility for random walk needs to be included.

3.2 Random walk

As has been mentioned before, it is not likely that each individual that alights from a train immediately walks into the right direction. Besides that, it is even less likely that any individual would not deviate from their direction: there are other people or obstacles such as benches, signs, food dispensers and bins to dodge. Furthermore, sleepiness or haziness could make people change direction. For those reasons an additional term needs to be added to the formula to include these random fluctuations in people's direction and speed. The formula for the position of an individual i now looks as follows:

$$d\mathbf{x}_i(t) = \mathbf{u}_i(\mathbf{x}_i(t))dt + \sigma d\mathbf{W}_i(t)$$

wherein σ incorporates the possibility of random walk and is set at 1 for this study.

$$d\mathbf{W}(t) = [d\mathbf{W}_x(t)\ d\mathbf{W}_y(t)]^T$$

is a vector Wiener process (also known as Brownian motion) with independent components $d\mathbf{W}_x(t)$ and $d\mathbf{W}_y(t)$ in $\mathbb{R}^2$ [5]. These components are both independent from past intervals of time over an interval dt and have a normal (or gaussian) distribution $d\mathbf{W}_x, d\mathbf{W}_y \sim \mathcal{N}(0, dt)$ [5][7].

3.3 Repulsive forces

People are not able to walk through each other. Furthermore, they mostly are not comfortable with bumping into other people while moving around. Every individual has a certain comfort-zone they like others to stay out of. For that reason repulsive forces need to be included into the model. If it happens that an individual j passes the threshold of the comfort-zone of individual i, then individual i will move away from individual j. The displacement that occurs consequently is denoted by $\mathbf{u}_{ij}(\mathbf{x}_i(t))dt$. However, it is not always true that there is only this one individual j in the vicinity of individual i. For that reason, the total displacement caused by repulsive forces is the sum of all the displacement caused by the individuals that pass the threshold of the comfort-zone of individual i. Adding this component to the formula results in

$$d\mathbf{x}_i(t) = \mathbf{u}_i(\mathbf{x}_i(t))dt + \sigma d\mathbf{W}_i(t) + \sum_{j \neq i} \mathbf{u}_{ij}(\mathbf{x}_i(t))dt.$$

In this study the comfort-zone is set to a width of 1 meter, equal to the diameter of the passengers themselves. Note that despite this comfort-zone it is still possible for a passenger to touch another passenger due to a great speed and a large time step.

3.4 Mechanotaxis

As has been mentioned before, there are mechanotactic processes on a platform. These are, for example, signs and announcements. This lessens the influence of any possible random walk component.

3.5 Final model

Putting everything that has been said so far together, the forces influencing one passenger on a platform have been sketched in Figure 3.2.

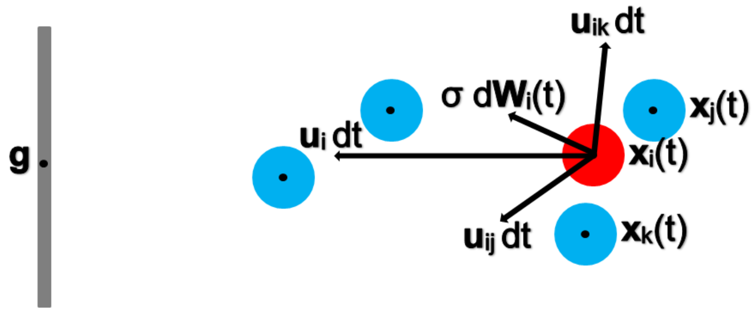

Figure 3.2: Goal $\mathbf{g}$. The red and blue circles are different individuals. The red circle, individual i, is the one for which the influencing forces are shown. Positions $\mathbf{x}_i(t)$ $\mathbf{x}_j(t)$ and $\mathbf{x}_k(t)$ of individual i, j and k respectively at time t, displacement $\mathbf{u}_i$ of individual i, random walk component $\sigma d\mathbf{W}_i(t)$ of individual i, and displacements $\mathbf{u}_{ij}$ and $\mathbf{u}_{ik}$ of individual i caused by the blue cells j and k, or individuals j and k, respectively.

In Figure 3.2 $\mathbf{g}$ is again the goal or the exit of the platform. The red circle, individual i, is the individual for which the forces that act upon it are shown. As has been explained, $\mathbf{u}_i\ dt$ is the displacement (speed and direction) of individual i at time t. This displacement is accompanied by a vector Wiener process $\sigma\ d\mathbf{W}_i(t)$ that slightly changes its course. Furthermore, the displacement is influenced by the surroundings. In this case it means that the positions $\mathbf{x}_j(t)$ and $\mathbf{x}_k(t)$ at time t of the blue circles j and k, or individuals j and k, respectively have breached the threshold of $\mathbf{x}_i(t)$ at time t. So the individual i will move away from individuals j and k. However, this 'moving away' will not be drastic, since individual i will still want to move towards the exit and not change their path too much. For the sake of clarity and simplification of the model, the magnitude of moving away has been assumed to be the same for every individual. Remember that everyone is assumed to have the same comfort zone. For that reason individuals j and k will also move away from individual i since individual i has passed the vicinity threshold of them as well as the other way around.

4 Python code explanation and elaboration

Each part of the mathematical model and how it was translated to Python code is discussed briefly. In the Python code the measurements are in decimeters, otherwise the platform and the individuals would have been too small to properly observe.

4.1 Chemotaxis

Since the exits of the platform can be regarded as sources of attraction to the passengers, the passengers have been given the correct direction from the start. This is their initial direction and it is where most passengers are going. Hence they 'follow the crowd'. However, other components, such as random walk and repulsive forces, will make them deviate from this path.

For the sake of completeness, we remind the reader that the chemotactic component in the displacement

$$\mathbf{u}(\mathbf{x}(t)) = \frac{\mathbf{g} - \mathbf{x}(t)}{\|\mathbf{g} - \mathbf{x}(t)\|} \cdot v_{walk}$$

has a normalized component. The position, i.e. the center, of each passenger is known. This position (x, y) can be obtained by calling the function 'canvas.coords(self.shape)' and adding 5 to both the x_1 and y_1 position as sketched in Figure 4.1.

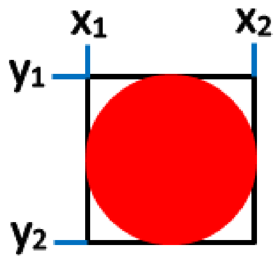

Figure 4.1: Visual representation of the coordinates of the red circle obtained by calling the function 'canvas.coords(self.shape)' in Python. This function returns a list of positions: $[x_1, y_1, x_2, y_2]$.

Calling 'pos = canvas.coords(self.shape)' returns a list of coordinates pos = $[x_1, y_1, x_2, y_2]$ as shown in Figure 4.1. So in the Python code, the coordinates (x, y) are $(pos[0]+5, pos[1]+5)$. Furthermore, the coordinates (g_x, g_y) of the goal are known. It then follows that $\mathbf{u}(\mathbf{x}(t)) = (g_x - x, g_y - y)/\|g - x\| \cdot v_{walk}$. Different cells have different speeds: some people walk faster than others or even run from one platform to the next to catch their train. For that reason, a 'multiplier' has been added to the model, randomly chosen from a given range. This multiplier ensures that each individual keeps their characteristic initial speed at all times. Their direction may change however due to random walk and interactions with their environment.

4.2 Random walk

As was mentioned earlier, the σ in $\sigma d\mathbf{W}_i(t)$ is set to 1 in this model for simplification. The random walk component is indeed a normal distribution with mean 0 and standard deviation 1, i.e. $d\mathbf{W}_x(t), d\mathbf{W}_y(t) \sim \mathcal{N}(0,1)$. Furthermore, passengers that enter the platform such that the distances to the exits of the platform are the same, take more time to choose a direction. They vary their paths a bit more and longer than passengers that are closer to one exit or the other. Besides that, so-called ‘green-zones’ have been added to the model. When individuals enter these zones the random walk component is left out since they are very close to the exit, hence they know exactly where to go to.

4.3 Repulsive forces

All passengers are put into one list: balls4. For each passenger in this list it is checked whether or not they are in the vicinity of another passenger and should therefore move away. Since this is checked for each passenger in the list the net movement can be seen as the sum $\sum_{j\neq i} \mathbf{u}_{ij}(\mathbf{x}_i(t))dt$.

4.4 Mechanotaxis

The fact that mechanotactic processes are present on a platform decreases the influence of the random walk component. For that reason, the random walk component in the Python code has a standard deviation of 1 and will therefore not have too big of an influence.

5 Results

The results can be roughly divided into two categories: qualitative and quantitative. The qualitative results, such as whether the individuals behave as they should, rely on the way the model has been programmed. I.e. it relies on how well the Python code works. The quantitative results are more about actual numbers: how much time does a certain number of people need to get off the platform? How many times are they in each other's vicinity? How many times do they maneuver around obstacles? These questions will be answered in this section, after some brief elaboration on the qualitative aspect of the model.

5.1 Qualitative results

During the study it was discovered that several improvements could be added to the model. Since this study is the first attempt to model the evacuation of a platform in this way, time constraints urged us to accomplish a limited amount of work. For now, the model that is presented in the appendix has been used.

The qualitative results are influenced by a couple of things:

- Each run is different since the model works with normal distributions for the speed and random walk component for each individual.
- New individuals appear whenever a certain object is clicked. One can try to keep the same amount of time between clicks, but since it is done manually one can never be absolutely sure it is always done the same way. This may influence the time the individuals need to exit the platform and how many individuals they cross paths with.
- At the beginning of each run the program runs fast and smoothly. However, the more individuals are added the slower it gets.
- Counting has been done with the human eye. Therefore, it might be the case that not everything was picked up. Nonetheless, the analyses have been done by the same person over a few runs which should include the same 'biases' everywhere, leveling the playing field. Future studies will use automated counting principles.

Despite the things listed above, the results that come hereafter can still be valued as legit since they all work with the same model and with that the same flaws. Furthermore, ten runs have been done and recorded for each individual part so that the results can be based on an average instead of only one or two observations.

5.2 Quantitative results

The questions posed at the beginning of section 5 (if relevant) will be answered in this subsection for three topics: one exit, benches as obstacles and an extra train. All three topics will be compared to a model in which there are two exits (left and right) and no obstacles, also referred to as the basic model. Figure 5.1 shows what this model looks like.

Figure 5.1: This is the basic model. Train exits: black rectangles at the top. Platform exits: left and right dark blue rectangles.

The black rectangles at the top of the platform are three exits from the train. This is where the individuals originate from. The blue rectangles at each side of the platform are the exits of the platform. Each individual chooses one of them as a goal. For now, it has been assumed that each individual chooses the closest exit as their goal. In future models, a randomizer can be used to enable individuals alighting from the left exit of the train to choose the right exit of the platform as their goal. The same goes for passengers originating from the right exit of the train choosing the left exit of the platform as their goal.

For every topic ten simulations were run for 45 people as well as 150 people on the platform. From these simulations the mean was calculated for each of the three questions (if relevant) to serve as a point of reference.

5.2.1 Basic model: two exits, no obstacles

As was said before, all topics will be compared to a basic model. Before these comparisons can take place, the results of the basic model need to be stated.

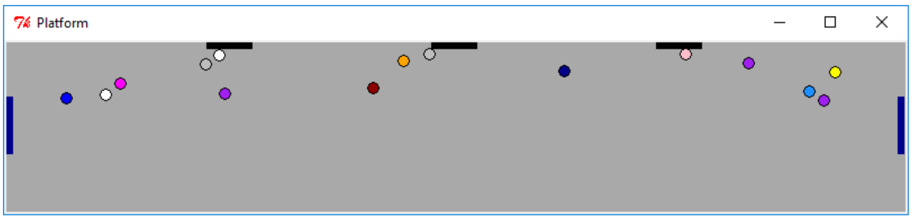

Figure 5.2: This is the basic model. Train exits: black rectangles at the top. Platform exits: left and right dark blue rectangles. Colored circles are the passengers.

Figure 5.2 shows a situation of the basic model in which individuals alighted from the train onto the platform. Note that the colors of the individuals differ. This was done on purpose to make it easier to distinguish between them and with that be able to observe their behavior better.

After performing ten runs of the basic model, it was found that 45 people needed only an average of 15 seconds to leave the platform. Furthermore, the average number of times there was contact between individuals is 12.5. With contact is meant that either the individuals touch or are in each other's vicinity such that they want to move away from each other.

Performing ten more runs with 150 people resulted in an average of 64.3 seconds needed for all individuals to leave the platform. The total average number of encounters between individuals is 86.2.

5.2.2 One exit

Intuitively, two exits is better than one since the passenger flows can be equally distributed among the exits and therefore relieve some pressure of the other one. It is interesting to see how much more contact there is between individuals when there is only one exit versus two exits. Figure 5.3 shows a situation with one exit.

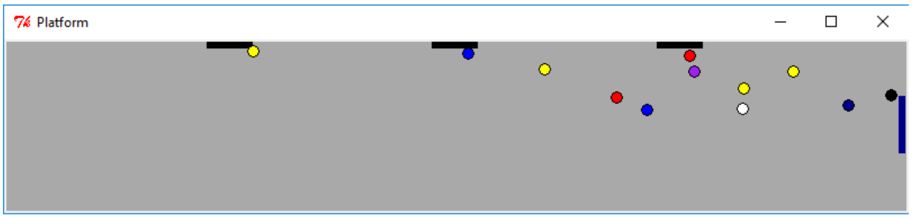

Figure 5.3: Train exits: black rectangles at the top. Platform exit: right dark blue rectangle. Colored circles are the passengers.

After analyzing ten runs of the model with one exit and 45 passengers, the following can be said: of those ten runs the average time that was needed for 45 people to enter and exit the platform is 21.2 seconds. Compared to the basic model that is an increase of 41.3%. That means that having more exits is beneficial for the flow of the passengers: they can all leave the platform faster. Furthermore, the average number of times there was contact between individuals is 34.6, an increase of 176.8%!. So not only are people able to leave the platform sooner, they also bump into each other much less. These two things together make the passenger flows on a platform more efficient and pleasant.

The same things can be said for 150 passengers on the platform, since the average time needed for them all to leave the platform increased by 31.7% to 84.7 seconds and the average number of encounters increased by 108.5% to 179.7.

5.2.3 Benches as obstacles

Every platform has at least one bench to sit on. In this model, it was assumed that there are three pairs of benches: three benches for each side of the platform. Five people can be seated on each bench, i.e. the benches are five times as long as the individuals but have the same width. See Figure 5.4 as an example.

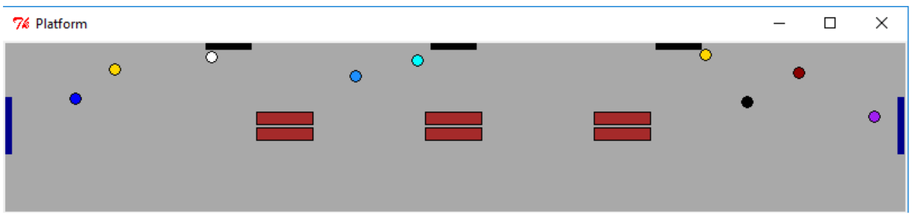

Figure 5.4: Three pairs of benches added to the platform as obstacles. Train exits: black rectangles at the top. Platform exits: left and right dark blue rectangles. Size of one bench: 50 dm by 10 dm, which is five times as long as one passenger and equally as wide. Colored circles are the passengers.

Analyzing the ten runs of the model with benches gave the following results: the average time needed for 45 passengers to leave the platform is 15.3 seconds. This is a small increase of 0.3 seconds, or 2%, compared to the basic model. The individuals were in contact with each other for an average of 12.6 times, an increase of 0.8%, and 3.1 times with the benches. Notice that, compared to the basic model, the number of times the individuals were in contact with each other has increased only ever so slightly. This increase of only 0.1 can be just due to using only ten runs but it might of course also be due to the addition of the benches to the model. The same goes for the time needed to leave the platform.

In regard to the time needed to leave the platform and the number of encounters between individuals, the results of this model with 150 passengers shows more promise: the average time needed for 150 individuals to leave the platform is 70.5 seconds, an increase of 9.6%. The average amount of contact between the individuals increased by 8.0% to 93.1. The contact with benches is 13.4 times. However, to be able to say more about whether benches have a (strong) influence on the flow of the passengers, the size of the benches was increased by 20%, 30% and 50%, both in length and in width. The model with the benches as shown in Figure 5.4 will from here on be referred to as the normal bench model. This model will be used as a point of reference for the number of encounters of individuals with benches, since the basic model itself does not include benches.

Benches: increased length and width by 20%
Figure 5.5 shows the normal bench model with 20% bigger benches, both in width and in length.

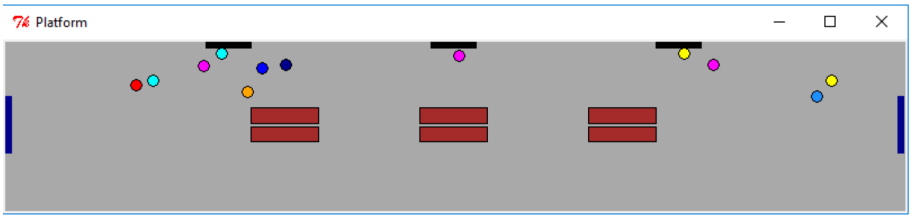

Figure 5.5: Three pairs of benches added to the platform as obstacles. Train exits: black rectangles at the top. Platform exits: left and right dark blue rectangles. Size of one bench: 60 dm by 12 dm, which is each 20% more than the original benches. Colored circles are the passengers.

The average time needed for 45 individuals to leave the platform is 15.9 seconds. Compared to the basic model that is an increase of 0.9 seconds, i.e. 6%. Furthermore, the contact between individuals and with the benches has increased by 0.5, i.e. 3.2%, compared to the basic model and 0.8 compared to the normal bench model. These may seem like small increases, but they are increases nonetheless.

Ten runs of this model with 150 passengers on the platform resulted in an average time of 73.5 seconds for them all to leave the platform. That is an increase of 14.3%. The number of encounters between individuals was measured to be 100.8, an increase of 16.9% compared to the basic model. Furthermore, compared to the normal bench model, the encounters of individuals with benches increased by 2.9 to 16.5 times.

Benches: increased length and width by 30%
Figure 5.6 shows the normal bench model with 30% bigger benches, both in width and in length.

Compared to the basic model, the average time needed for 45 passengers to leave the platform increased by 1.4 seconds, i.e. 9.3%. The contact between the individuals increased by 2.5 to 15, an increase of 20%. The encounters of individuals with benches is an average of 4.1 times, an increase of 1.0 compared to the normal bench model.

For 150 passengers on the platform, the average time measured for them all to leave the platform is 74.9 seconds, an increase of 16.5%. Compared to the basic model, the number of times there

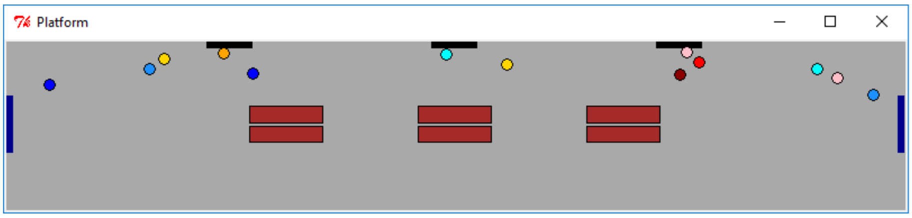

Figure 5.6: Three pairs of benches added to the platform as obstacles. Train exits: black rectangles at the top. Platform exits: left and right dark blue rectangles. Size of one bench: 65 dm by 13 dm, which is each 30% more than the original benches. Colored circles are the passengers.

was contact between individuals increased by 36.5% to an average of 117.7 times. The contact of individuals with the benches increased by 3.7 to 17.1 times.

Benches: increased length and width by 50%

Figure 5.7 shows the normal bench model with 50% bigger benches, both in width and in length.

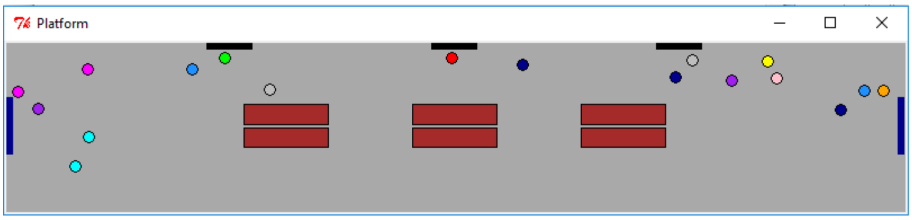

Figure 5.7: Three pairs of benches added to the platform as obstacles. Train exits: black rectangles at the top. Platform exits: left and right dark blue rectangles. Size of one bench: 75 dm by 15 dm, which is each 50% more than the original benches. Colored circles are the passengers.

An average increase of 2.4 seconds was witnessed for the time needed for 45 individuals to leave the platform. That is an increase of 16%. Furthermore, the contact between individuals increased by 5.8, i.e. 46.4%. The contact of individuals with the benches increased to an average of 6 times.

On average 150 passengers needed 78.8 seconds to leave the platform, an increase of 22.6% compared to the basic model. Furthermore, the number of times individuals had contact with each other averages to 121.8 times. That is an increasement of 41.3%. Compared to the normal bench model, the average number of times individuals were in contact with benches is 20.9 times.

It now seems fair to say that adding benches to the platform is not beneficial for the overall passenger flow in terms of time needed to leave the platform and the number of times individuals are in contact with each other or the benches. It can also be concluded that the bigger the benches, the more the contact with not only the benches themselves, but also with other individuals. That observation insinuates that having benches on the platform increases the contact between individuals. This is not desired. However, it can be argued that not adding benches at all would not create a good atmosphere on the platform. People want a place to sit down if they like. Therefore, and also due to the small increase of contact between people when benches were added in the first place, it is still advisable to place benches on a platform. It should nonetheless be kept in mind not to go overboard with the size of the benches.

5.2.4 Extra train

It can happen that two trains arrive at the same platform at around the same time. In that case, passengers will flow from both trains onto the platform at around the same time. For this model twice as many individuals entered the platform and it was assumed that they all want to leave the platform and will not board the other train. However, this does not mean that it will take twice as long for them to leave the platform, since multiple people can simultaneously walk through the exit. Still, intuitively having two trains at the same platform at the same time will be very unfavorable for the flow of the passengers: it is likely that there will be more contact between individuals and it will probably still take more time for everyone to leave the platform. These hypotheses were tested by running the model with an extra train ten times. Again, 45 people or 150 people entered the platform from the top train. However, for every person that entered the platform from the top train another passenger from the bottom train simultaneously entered the platform. That means that the total amount of passengers that entered the platform was 90 or 300. The results are stated below Figure 5.8.

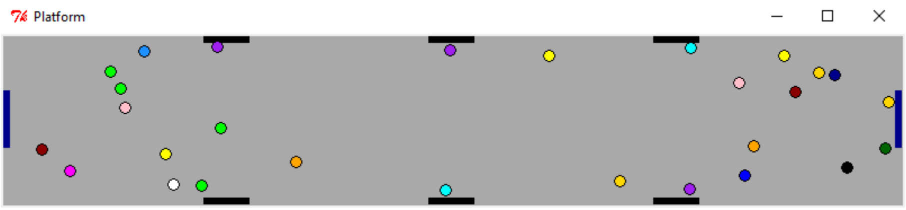

Figure 5.8: Extra train added to the platform. Train exits: black rectangles. Platform exits: left and right dark blue rectangles. Colored circles are the passengers.

The average time needed for the 90 individuals to leave the platform was measured to be 22.1 seconds. Compared to the basic model that is an increase of 7.1 seconds, i.e. 47.3 %. So almost half as long. Furthermore, the contact between individuals on average increased to 38.2 times. That is an increase of 205.6%!

For the total of 300 passengers the average time needed to leave the platform is 113.2 seconds. Compared to the basic model that is an increase of 76.0%. The average number of encounters between individuals increased by 120.3% to 189.8 times.

Based on the results stated above it can be said that it is tremendously unfavorable to have two trains arrive at the same platform at around the same time. Especially in terms of contact between individuals. Of course, only the extreme version has been tested so far. More research needs to be done for situations in which, for example, only half as many individuals exit from one train compared to the other.

5.3 Graphical overview of the results

Several graphs were made to put all that has been said in Subsection 5.2 into perspective. Each of these graphs provide an answer to one of the questions posed at the beginning of Section 5.

Figures 5.9 and 5.10 respectively show the results for the time needed for 45 and 150 individuals to enter and leave the platform. From these figures it can easily be concluded that the basic model is the most ideal situation and the situations with one exit and an additional train are the worst. Furthermore, benches seem to have only a minor effect on the time needed for all passengers to leave the platform. Besides that, the relative increase for the extra train model in Figure 5.10 stands out. This high value is partly caused by computational limitations that led the program to slow down tremendously.

Figures 5.11 and 5.12 show the number of encounters between 45 and 150 individuals on the platform respectively. Again, the situations with one exit and with an extra train stand out as the worst ones. The models with the benches show an increase in encounters as the size of the benches increases. The effect that adding benches in the first place has is almost negligible, especially in the case of 45 people on the platform. Therefore, it can once more be said that adding benches to the platform is advisable. Not only to improve upon the comfort of the passengers, but also since the effect on the passenger flows is minimal. However, as Figures 5.11 and 5.12 show, the benches should not become too big.

Figures 5.13 and 5.14 support the statement made previously: the bigger the benches the more encounters with the benches themselves and between the individuals. Based on Figures 5.13 and 5.14 it seems likely that encounters with the benches result in encounters between individuals. It should therefore be stressed again to not go overboard with the sizes of the benches.

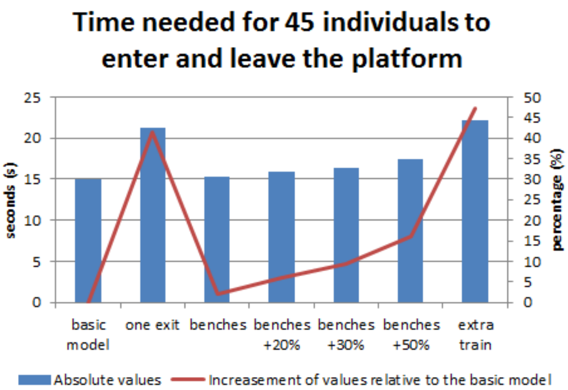

Figure 5.9: Time in seconds needed for 45 people to enter and leave the platform. The blue bars correspond with the left vertical axis and represent the total amount of seconds. The red line corresponds with the right vertical axis and represents the increasement of the time needed for 45 individuals to leave the platform relative to the basic model.

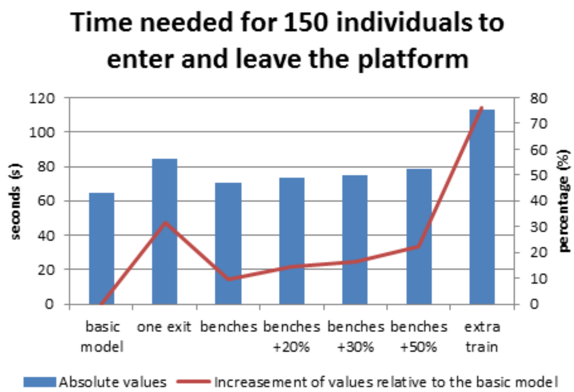

Figure 5.10: Time in seconds needed for 150 people to enter and leave the platform. The blue bars correspond with the left vertical axis and represent the total amount of seconds. The red line corresponds with the right vertical axis and represents the increasement of the time needed for 45 individuals to leave the platform relative to the basic model.

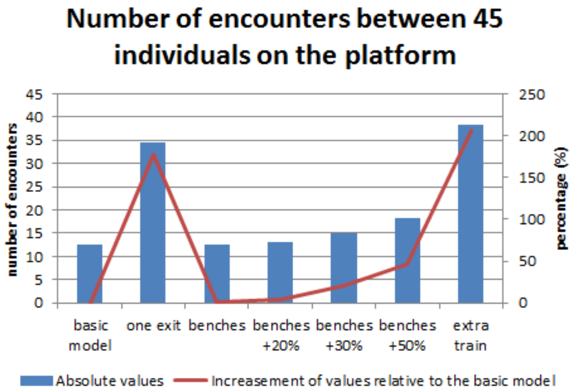

Figure 5.11: Number of encounters between 45 individuals on the platform. The blue bars correspond with the left vertical axis and represent the total number of encounters between individuals. The red line corresponds with the right vertical axis and represents the increasement of the encounters between 45 individuals relative to the basic model.

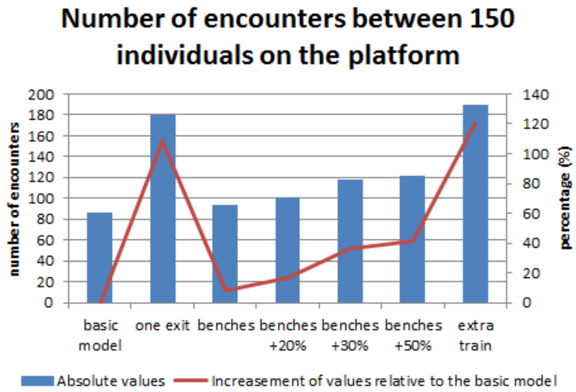

Figure 5.12: Number of encounters between 150 individuals on the platform. The blue bars correspond with the left vertical axis and represent the total number of encounters between individuals. The red line corresponds with the right vertical axis and represents the increasement of the encounters between 150 individuals relative to the basic model.

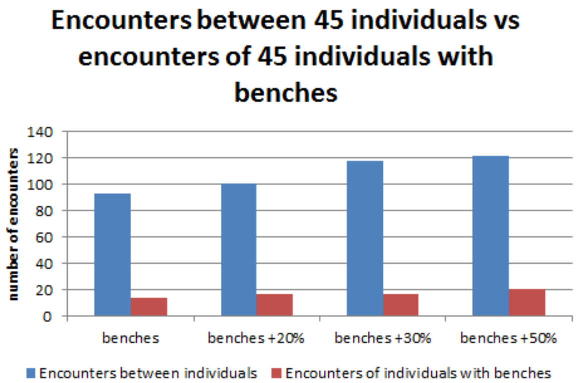

Figure 5.13: Number of encounters between 45 individuals vs encounters of 45 individuals with benches. The blue bars represent the encounters between individuals. The red bars represent the encounters of the individuals with the benches.

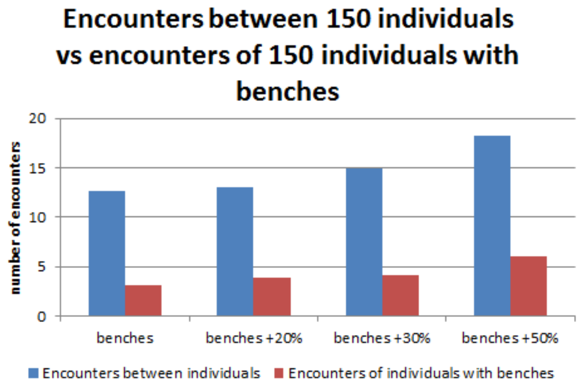

Figure 5.14: Number of encounters between 150 individuals vs encounters of 150 individuals with benches. The blue bars represent the encounters between individuals. The red bars represent the encounters of the individuals with the benches.

6 Discussion

This report marks the beginning of looking into passenger flows on a platform. However, there is still a lot that can be done to improve upon the model provided in this report. Given the time restriction of only seven weeks, only a second train and benches have been added to the model thus far. As for the train, it only has been looked at what the passenger flows are when both trains arrive around the same time. This does not happen often in reality. Still, it has been shown that having two trains arrive around the same time is not beneficial for the passenger flows. Furthermore, benches are not the only obstacles that can be found on a platform. There are also waste bins, food dispensers, a kiosk, smoking areas, etc. These all provide a different kind of obstacle and can serve as an addition to the model. Also, besides looking at the variance of bench-sizes, the size of the exits could also have been varied. Intuitively, the bigger the exit the better the flow of passengers and the lesser the contact between them.

The model presented in this report does not take people with another goal than getting out of the train and leaving the platform into account. However, in reality other people mostly want to enter the train as well, creating another obstacle. Also, some people might need to transfer to a train that arrives at exactly the same platform, so they will not be going anywhere. They might stand still and be an obstacle or they could use a bench to sit down on. This could be taken into account in future studies on passenger flows on train platforms.

Not everyone travels alone. It could be possible to model a group of friends moving along together as a cluster. A possible way for defining a cluster is through mechanotaxis: the movement of one of the group triggers the movement of the whole group. Defining a cluster, however, is complex, but when it has been done perhaps a Poisson distribution can be used to have a group of people come out of the train together now and then. Besides that, when clusters are defined in the model they can be used to incorporate a so-called 'jam-problem'. If there are many people in front of you, then you need to slow down your pace as to not run into them. It is also possible to make individuals move in between clusters of passengers. As has been mentioned before, this is rather complex to achieve but would make the model be more in line with reality.

Size is another thing that needs to be discussed on several levels. First of all, people come in many shapes and sizes. The model presented in this report assumed everyone to have the exact same size. This is not true in reality. Furthermore, the size of the platform itself is not representative of reality either. However, these dimensions have been chosen to simplify the model, shorten running time and to be able to see the behavior of the passengers more clearly. Even so, the observed effects as discussed in section 5 are still legit since they have all taken place on the same platform (in dimensions) and have all been compared to the same basic model.

The model used in this report is based on the Particle Model. When looking at the components of the Particle Model and translating it to passenger flows on a platform it does seem like a good fit. However, there are alternative models out there that could provide a fit that is equally as good or even better. Even with that being said, the model provided in this report can serve as a basis. Not only for additions to a platform, but also for other places that deal with flows of people. Examples of these other places are a concert hall, a festival, a school canteen or a restaurant. How long will it take for everyone to get out during a terrorist attack or if the lights go out and everyone's movements consist mainly of random walk during panic? These are all situations that can be modeled using the model of this report as a basis.

Apart from the model in which a train was added, this study has only looked at simulations with 45 and 150 passengers on the platform. For future studies it is preferable to increase the number of passenger to, for example, 500 passengers. This will probably show clearer dependencies. For the model presented in this report this is not doable: the platform itself is too small and the program became very slow with only 150 passengers on it already.

This study has not taken confidence intervals into account. These intervals could provide more information about whether the observed decreases and increases are statistically significant, i.e. whether they lie in the 95% confidence interval or not. In the model used for this study, a stochastic term is introduced in the migration of the individuals. This stochastic term is treated with the help of a Wiener process, which itself is based on the normal distribution. Assuming that there are no other individuals or obstacles on the platform, the displacement of an individual over a time interval will be distributed normally as well. In this normal distribution the mean position follows from the deterministic component, or chemotactic component in this case, and the variance will be $\sigma^2\, dt$. The term σ refers to the σ of the random walk component. However, there actually are many other individuals on the platform that can come in (direct) contact with each other. Because of this, their behavior will become more chaotic, which in turn justifies the question whether or not a normal distribution is still applicable to the results. This is an important matter since one needs to know whether differences in the outcomes are significant from a statistical point of view. Normality could be investigated by the use of the Anderson-Darling test for instance. That is why the simulations need to be repeated several times. In this study, all simulations were repeated ten times. In fact, it is a Monte-Carlo simulation [8] that gives convergence of the order $\mathcal{O}(n^{-1/2})$, where n represents the number of observations or in this case the number of simulations. This convergence is not particularly fast, but for large n the Central Limit Theorem from statistics gives convergence to a normal distribution. In combination with a test such as the Anderson-Darling test, the Central Limit Theorem justifies using the normal distribution for the interpretation of the results, e.g. evacuation time or number of encounters with individuals or benches. Note, however, that the number of simulations done in this study is not enough to justify a normal distribution yet. The number of simulations per case was low because of computational limitations.
Now that it has been justified that confidence intervals should be used for investigating whether changes are statistically significant, one can use the following interval of confidence

$$x_s - \frac{t_{a/2}(n) \cdot s}{\sqrt{n}} < \mu < x_s + \frac{t_{a/2}(n) \cdot s}{\sqrt{n}}$$

in which x_s is the sample average, n the number of observations or simulations, s the sample standard deviation, and $t_{a/2}(n) = t_{0.025}(n)$ the student's t-distribution for the probability value. Note that $a/2 = 0.025$ since we are referring to 95% confidence intervals, which means there is 5% remaining for the left and right side of the normal distribution in total. As mentioned earlier, this confidence interval can be used to test whether two populations differ significantly. In this context the populations refer to the different situations that were modeled, such as one exit, benches as obstacles and an additional train.

7 Conclusions

The model introduced in this report uses chemotaxis, random walk, repulsive forces and mechanotaxis as a basis for modeling passenger flows on a platform. The goal was to gain insight in the quantitative results concerning time needed for a certain amount of passengers to leave the platform, the number of encounters between individuals and the number of encounters of individuals with benches.

The results indicate that multiple exits are beneficial for the flow of the passengers. Furthermore, adding benches of size 50 dm by 10 dm does not have a great effect on either of the researched quantitative topics, unless the size of the benches is increased above 20%. Moreover, having another train enter the same platform as another train at around the same time is tremendously unfavorable for the passenger flows.

The model presented in this report can be used as a basis for future studies about passenger flows on a platform or anywhere else where flows of people are applicable, such as concert halls, festivals, restaurants and canteens.

References

[1] NS Stations. Facts & Figures. Retrieved from: http://www.nsstations.nl/ns-stations/facts-figures.html

[2] M. Dudaie, D.Weihs, F.J. Vermolen, A.Gefen. (2015). Modeling migration in cell colonies in two and three dimensional substrates with varying stiffnesses. *In Silico Cell and Tissue Science*. Retrieved from: https://link.springer.com/article/10.1186/s40482-015-0005-9

[3] F.J. Vermolen. (2015). Particle methods to solve modelling problems in wound healing and tumor growth. *Computational Particle Mechanics*, 2(4). Retrieved from: http://repository.tudelft.nl/islandora/object/uuid:3b37b4a2-dcff-4885-9f78-a095e21345ad?collection=research

[4] E.F. Keller. (1971). Model for Chemotaxis.]emphJ. theor. Biol., 30, 225-234. Retrieved from: http://jxshix.people.wm.edu/2013-taiwan/Keller-Segel-1971-JTB.pdf

[5] F.J. Vermolen, E.C.M.M. Arkesteijn, A. Gefen. (2016). Modeling the Immune System Repsonse to Epithelial Wound Infections. *Journal of Theoretical Biology*, 393, 158-169. Retrieved from: http://www.sciencedirect.com/science/article/pii/S0022519316000370

[6] T. Kawano, S. Kidoaki. (2011). Elasticity boundary conditions required for cell mechanotaxis on microelastically-patterned gels. *Biomaterials*, 32(11). Retrieved from: https://www.ncbi.nlm.nih.gov/pubmed/21276611

[7] T.Hida. (1980). Brownian motion. *Applications of Mathematics*, 11, 45. Retrieved from: https://link.springer.com/chapter/10.1007/978-1-4612-6030-1_2

[8] Wikipedia. (June 2017). Monte Carlo method. Retrieved from: https://en.wikipedia.org/wiki/Monte_Carlo_method

[9] C. Vuik, F.J. Vermolen, M.B. van Gijzen, M.J. Vuik. (2016). *Numerical Methods for Ordinary Differential Equations*. Delft: Delft Academic Press.

[10] A.B. Downey. (2012). *Think Python*. Sebastopol: O'Reilly Media Inc, 217-228.

APPENDIX

Some lines in the Python code are too long to fit on one line. Those have been cut in two to fit into the frame. The symbol '&&' shows that this has been done to a line. Mind that the next line, also denoted by '&&', should then in fact be 'glued' to the previous line in order for the code to work properly, should you wish to try it. Mind that the code added here is the one that belongs to the basic code as explained in section 5.

Python Code

```
import Tkinter
import sys
import random
import time
import numpy as np
from threading import Timer
from math import *


#left exit train#
class Ball1:
    def __init__(self, color):
        self.shape = canvas.create_oval(195, 5, 205, 15, fill=color)
        self.xspeed = -WIDTH/(2*HEIGHT)
        self.yspeed = 1.
        self.multiplier = random.uniform(0.5,2.9)

    def move(self):
        canvas.move(self.shape, self.xspeed, self.yspeed)
        self.xspeed = -WIDTH/(2*HEIGHT)
        self.xspeed *= self.multiplier
        self.yspeed = 1.
        pos = canvas.coords(self.shape)
        eps = 15
        #non-green zone, with random walk#
        if pos[0]>75 or pos[0] < 725:
            self.xspeed += np.random.normal(0,1)
            self.yspeed += np.random.normal(0,1)
        #green zones#
        if pos[0]<=75 or pos[0] >= 725:
            if abs(pos[1]+5 - 75) <= eps:
                self.yspeed = 0
            return
            if pos[1] + 5 < 60:
                self.xspeed = -(pos[0]+5)/(pos[1]-75)
                self.xspeed *= self.multiplier
                self.yspeed = - 1
            return
            if pos[1] + 5 > 90:
```

```
                self.xspeed = -(pos[0]+5)/abs(pos[1]-75)
                self.xspeed *= self.multiplier
                self.yspeed = 1
            return

        #right exit#
        if pos[2] >= WIDTH and pos[1] <= 50 or pos[2] >= WIDTH and &&
        &&pos[3] >= 100:
            self.xspeed = -self.xspeed
        elif pos[2] >= WIDTH and pos[1] > 50 and pos[3] < 100:
            balls4.remove(self)
            canvas.delete(self.shape)
        #left exit#
        elif pos[0] <= 0 and pos[1] <= 50 or pos[0] <= 0 and pos[3] >= 100:
            self.xspeed = -self.xspeed
        elif pos[0] <= 0 and pos[1] > 50 and pos[3] < 100:
            balls4.remove(self)
            canvas.delete(self.shape)
        #upper- and lowerbound#
        elif pos[3] >= HEIGHT or pos[1] <= 0:
            self.yspeed = -self.yspeed

#middle exit train#
class Ball2:
    def __init__(self, color):
        self.shape = canvas.create_oval(395, 5, 405, 15, fill=color)
        self.multiplier = random.uniform(0.5,2.9)
        x = random.randint(1,2)
        if x == 1:
            self.xspeed = WIDTH/HEIGHT
            self.yspeed = 1.
        if x == 2:
            self.xspeed = -WIDTH/HEIGHT
            self.yspeed = 1.

    def move(self):
        canvas.move(self.shape, self.xspeed, self.yspeed)
        pos = canvas.coords(self.shape)
        eps = 15
        if 385<pos[0]<405:
            x = random.randint(1,2)
            if x == 1:
                self.xspeed = WIDTH/HEIGHT
                self.xspeed *= self.multiplier
                self.yspeed = 1.
                self.xspeed += np.random.normal(0,1)
                self.yspeed += np.random.normal(0,1)
            if x == 2:
```

```
            self.xspeed = -WIDTH/HEIGHT
            self.xspeed *= self.multiplier
            self.yspeed = 1.
            self.xspeed += np.random.normal(0,1)
            self.yspeed += np.random.normal(0,1)
        return
    #non-green zone, with random walk, left exit#
    elif 75<pos[0]<=385:
        self.xspeed = -WIDTH/HEIGHT
        self.xspeed *= self.multiplier
        self.yspeed = 1.
        self.xspeed += np.random.normal(0,1)
        self.yspeed += np.random.normal(0,1)
        return
    #green zones, without random walk, left exit#
    elif pos[0]<=75:
        if abs(pos[1]+5 - 75) <= eps:
            self.yspeed = 0
        return
        if pos[1] + 5 < 60:
            self.xspeed = (pos[0]+5)/(pos[1]-75)
            self.xspeed *= self.multiplier
            self.yspeed = 1
        return
        if pos[1] + 5 > 90:
            self.xspeed = (pos[0]+5)/abs(pos[1]-75)
            self.xspeed *= self.multiplier
            self.yspeed = -1
        return
    #non-green zone, with random walk, right exit#
    elif 405<=pos[0]<725:
        self.xspeed = WIDTH/HEIGHT
        self.xspeed *= self.multiplier
        self.yspeed = 1.
        self.xspeed += np.random.normal(0,1)
        self.yspeed += np.random.normal(0,1)
        return
    #green zones, without random walk, right exit#
    elif 725<=pos[0]:
        if abs(pos[1]+5 - 75) <= eps:
            self.yspeed = 0
            return
        if pos[1] + 5 < 60:
            self.xspeed = (pos[2]+5 - 800)/(pos[3]-75)
            self.xspeed *= self.multiplier
            self.yspeed = 1
            return
        if pos[1] + 5 > 90:
```

```
                self.xspeed = (pos[2]+5-800)/abs(pos[3]-75)
                self.xspeed *= self.multiplier
                self.yspeed = -1
                return
        #right exit#
        if pos[2] >= WIDTH and pos[1] <= 50 or pos[2] >= WIDTH and &&
        &&pos[3] >= 100:
            self.xspeed = -self.xspeed
        elif pos[2] >= WIDTH and pos[1] > 50 and pos[3] < 100:
            balls4.remove(self)
            canvas.delete(self.shape)
        #left exit#
        elif pos[0] <= 0 and pos[1] <= 50 or pos[0] <= 0 and pos[3] >= 100:
            self.xspeed = -self.xspeed
        elif pos[0] <= 0 and pos[1] > 50 and pos[3] < 100:
            balls4.remove(self)
            canvas.delete(self.shape)
        #upper- and lowerbound#
        elif pos[3] >= HEIGHT or pos[1] <= 0:
            self.yspeed = -self.yspeed

#right exit train#
class Ball3:
    def __init__(self, color):
        self.shape = canvas.create_oval(595, 5, 605, 15, fill=color)
        self.xspeed = WIDTH/(2*HEIGHT)
        self.yspeed = 1.
        self.multiplier = random.uniform(0.5,2.9)

    def move(self):
        canvas.move(self.shape, self.xspeed, self.yspeed)
        self.xspeed = WIDTH/(2*HEIGHT)
        self.xspeed *= self.multiplier
        self.yspeed = 1.
        pos = canvas.coords(self.shape)
        eps = 15
        #non-green zone, with random walk#
        if 75 < pos[0]<725:
            self.xspeed += np.random.normal(0,1)
            self.yspeed += np.random.normal(0,1)
            return
        #green zones, without random walk#
        elif pos[0]>=725 or pos[0] <= 75:
            if abs(pos[1]+5 - 75) <= eps:
                self.yspeed = 0
            if pos[1] + 5 < 60:
                self.xspeed = (800-pos[0]+5)/(75-pos[1])
                self.xspeed *= self.multiplier
```

```
                self.yspeed = 1
            if pos[1] + 5 > 90:
                self.xspeed = (800-pos[0]+5)/abs(75-pos[1])
                self.xspeed *= self.multiplier
                self.yspeed = -1
            return
        #right exit#
        if pos[2] >= WIDTH and pos[1] <= 50 or pos[2] >= WIDTH and &&
        &&pos[3] >= 100:
            self.xspeed = -self.xspeed
            return
        elif pos[2] >= WIDTH and pos[1] > 50 and pos[3] < 100:
            balls4.remove(self)
            canvas.delete(self.shape)
            return
        #left exit#
        elif pos[0] <= 0 and pos[1] <= 50 or pos[0] <= 0 and pos[3] >= 100:
            self.xspeed = -self.xspeed
            return
        elif pos[0] <= 0 and pos[1] > 50 and pos[3] < 100:
            balls4.remove(self)
            canvas.delete(self.shape)
            return
        #upper- and lowerbound#
        elif pos[3] >= HEIGHT or pos[1] <= 0:
            self.yspeed = -self.yspeed
            return


def leftClick1(event):
    ball1 = Ball1(random.choice(colors))
    ball1.move()
    ball2 = Ball2(random.choice(colors))
    ball2.move()
    ball3 = Ball3(random.choice(colors))
    ball3.move()

    balls4.append(ball1)
    balls4.append(ball2)
    balls4.append(ball3)
    time.sleep(0.05)
    for ball in balls4:
        ball.move()
    root.update()



#reaction to other balls#
```

```
def react(ball):
    pos1 = canvas.coords(ball.shape)
    eps = 7
    mid1 = [(pos1[0]+pos1[2])/2,(pos1[1]+pos1[3])/2]
    for bal in balls4:
        pos2 = canvas.coords(bal.shape)
        if pos1[0]>=0 and pos1[1]>=0 and pos2[0]>=0 and pos2[1]>=0 or &&
        &&pos1[0]<=795 and pos1[1]<=145 and pos2[0]<=795 and pos2[1]<=145:
            mid2 = [(pos2[0]+pos2[2])/2,(pos2[1]+pos2[3])/2]
            if hypot(mid1[0]-mid2[0], mid1[1]-mid2[1]) <= 10. + eps:
                ball.xspeed += 1
                bal.xspeed -= 1
                ball.yspeed += 1
                bal.yspeed -= 1
                ball.move()
                bal.move()


def close_window():
    root.destroy()
    sys.exit()

root = Tkinter.Tk()
root.protocol("WM_DELETE_WINDOW", close_window)
WIDTH = 800.
HEIGHT = 150.

#platform#
root.title("Platform")
canvas = Tkinter.Canvas(root, width=WIDTH, height=HEIGHT, bg='darkgrey')
#platform exits#
canvas.create_rectangle(0,50,7,100,fill='darkblue',outline='darkblue',&&
&&width=1)
canvas.create_rectangle(795,50,800,100,fill='darkblue',outline='darkblue',&&
&&width=1)

#all balls (aka people)#
balls4 = []
colors = ['red', 'green', 'blue', 'orange', 'yellow', 'cyan', 'magenta',
          'dodgerblue', 'grey', 'gold', 'pink', 'purple',
          'black', 'white', 'darkblue', 'darkred', 'darkgreen']

#exits train#
obj1Id = canvas.create_rectangle(380,0,420,5,fill='black',outline='black',&&
&&tags="obj1Tag")
canvas.create_rectangle(180,0,220,5,fill='black',outline='black')
canvas.create_rectangle(580,0,620,5,fill='black',outline='black')
canvas.tag_bind(obj1Id, '<ButtonPress-1>', leftClick1)
```

```
canvas.pack()

check = True
while check:
    for ball in balls4:
        try:
            react(ball)
        except:
            continue
    time.sleep(0.05)
    try:
        root.update()
    except:
        check = False



root.mainloop()
```