Health care system

Università degli Studi di Milano
Dipartimento di Informatica
Course "Simulation" – Exam Project, A.Y. 2023-2024
Student: **Sara Cinquini**

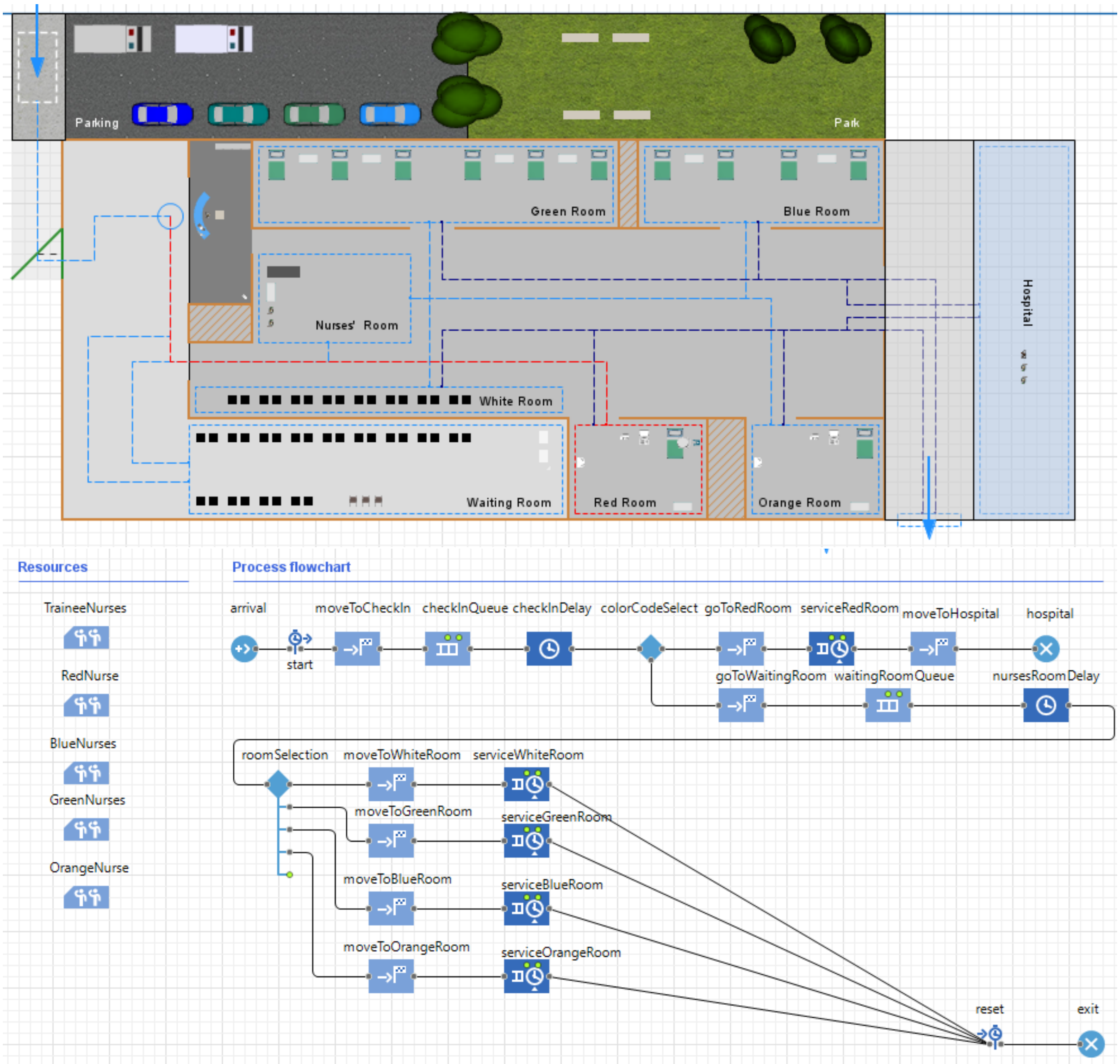
CONTENTS

1 INTRODUCTION

In the healthcare sector, efficient management of resources and patients is a constant challenge, especially in the context of the emergency room. The goal of this project is to analyse the flow and management of patients in a **hospital system**, focusing on key elements such as patient influx, queue management, resource allocation (nurses and operating rooms) and waiting times.

To simulate this system, I've decided to use **AnyLogic**, a powerful simulation software that allows modelling complex dynamics. The project is based on the observation of the emergency room of the *San Donato Milanese* hospital, trying to replicate its daily functioning. The simulation, structured on a <u>time basis in minutes</u>, is designed to represent a typical 24-hour working day, at the end of which the system stops.

Through this simulation, we aim to understand the functioning of an **emergency room** and to explore possible improvements that could reduce waiting times and optimize resource allocation, as improving the efficiency of the healthcare system.



1.1 THE MODEL

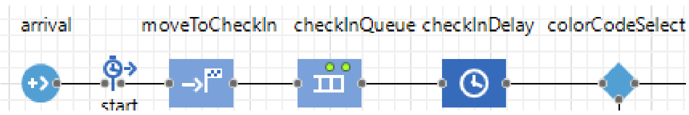
For this scenario, it was chosen to use a **discrete event-based** and **agent-based** simulation model, since the dynamic interactions between agents (patients and nurses) and the resources available within the emergency room will be simulated.

2 IMPLEMENTATION

2.1 WORKING FLOW'S EXPLANATION

The emergency room process is structured as follows: a patient arrives in the emergency room and must present himself at the reception desk where a technician or nurse will assess the patient's condition and assign him a colour code (assignment using probabilities). In the hospital setting, the codes are as follows (*ordered from the highest to the lowest code*):

- **red** code, represents the most complex situation with the highest priority, therefore defined as <u>maximum emergency</u>;
- **orange** code, the patient's conditions are certainly not simple, but more manageable; a fairly high level of attention is certainly required, for this reason it is defined as <u>urgency</u>;
- **blue** code, the patient does not have serious injuries for which time plays an important role; requires treatment even with longer timeframes, for this reason it is defined as <u>deferrable urgency</u>;
- **green** code, the situation does not require timely treatment, for this reason it is defined as <u>minor urgency</u>;
- **white** code, these are simple situations that can also be managed by specialists/trainees, for this reason it is defined as <u>non-urgency.</u>



2.1.1 Time distributions

Patient arrivals follow an **exponential distribution**, while treatment times are distributed according to **a triangular distribution** based on patient severity.

The exponential distribution was chosen because it realistically represents the <u>random nature of patient arrivals</u> in an emergency department. Patients do not arrive in a regular and predictable way, but rather irregularly, and the exponential distribution allows this variability to be simulated.

2.1.2 Code for probabilistic assignment of colour codes

```
double random = uniform(0, 1);
if (random <= 0.1) {
      person3D.setColor("Material__5__Surf", red);
      color_code = 0;
} else if (random <= 0.3) {
      person3D.setColor("Material__5__Surf", orange);
      color_code = 1;
} else if (random <= 0.6) {
    person3D.setColor("Material__5__Surf", blue);
    color_code = 2;
} else if (random <= 0.9) {
      person3D.setColor("Material__5__Surf", green);
      color_code = 3;
} else {
      person3D.setColor("Material__5__Surf", white);
      color_code = 4;
}
```

This approach uses **cumulative probabilities**. Each bin is set so that the sum of the previous probabilities determines the lower bound for each subsequent colour.

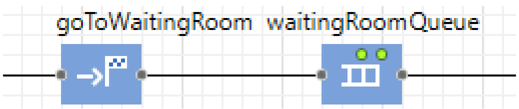
The uniform(0, 1) function generates a random number with uniform distribution over [0,1], meaning that each value in that bin has an equal chance of being generated.
The code uses a simple form of cumulative distribution, where the sum of the probability bins covers the entire interval [0,1].

After the colour code has been assigned, the patient will sit in the **waiting room**, where he/she will wait to be called. A **priority algorithm** is used to manage the queue, such that depending on the colour code and the waiting time, the patient will be assigned a priority value.

The formula that has been hypothesized is the following:

$$priority = pt_{base} + increment_t$$

Where:
- pt_{base}, it is the score initially assigned based on the colour code.
- $increment_t$, it is the score that increases over time, based on the minutes of waiting.



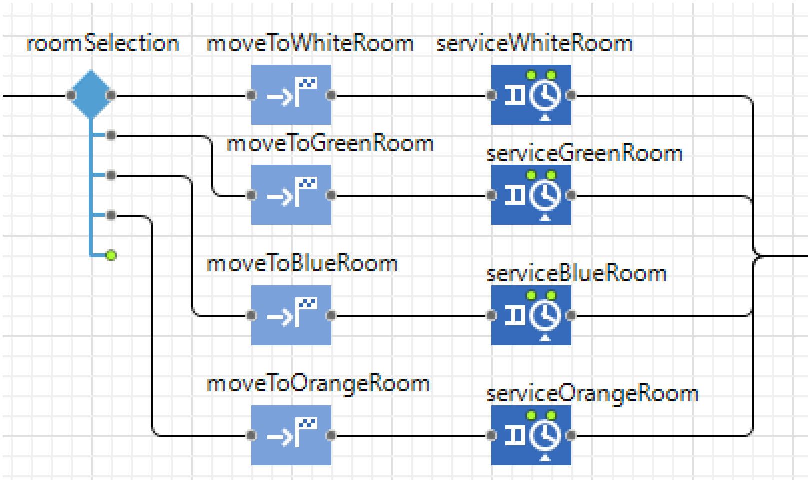
When the patient is called, he/she will move from the waiting room to the desk where the nurses are present, who will indicate the correct room in which he/she will be placed.
In the emergency room there are 5 different rooms, one for each colour code. The capacity depends on the availability of resources in the Emergency Room, in our case:
- The red room has a maximum capacity of 1 patient.
- The orange room has a maximum capacity of 1 patient.

- The blue room has a maximum capacity of 4 patients.
- The green room has a maximum capacity of 6 patients.
- The white room has a maximum capacity of 15 patients.

Whenever a patient is correctly assigned to a room, he or she is automatically placed in a queue (in FIFO mode), so he or she will have to wait his or her turn before being treated by the ward nurse.

Each room is associated with a **number of nurses** who will dedicate themselves to the patients present. Since the codes with higher priority need prompt assistance, it was decided to divide the team as follows:

- For orange and red codes, for which the operating room can be occupied by one patient at a time, one nurse is assigned per ward respectively; this allows treatments to be speeded up.
- For blue codes, there can be a variable number of nurses ranging from 1 to 4.
- For green codes, it was decided to assign a maximum of 3 nurses.
- For white codes, the maximum number of residents is 2.



Once the patient has been properly examined and treated, it must be understood whether the problem has been managed or whether it is necessary to keep the patient for longer. Generally, those who hold a code equal to **green** or **white**, will end their stay in the emergency room and will be able to go to the exit. If, on the other hand, patients have **more urgent conditions** (therefore there is a probability of a possible complication), a further examination will have to be performed to determine whether it is necessary to move them to the appropriate hospital. Very urgent patients (i.e. those who have been assigned a red colour code) will certainly be moved to the hospital.

2.1.3 Needing more assistance

```
double probability = 0;

if (color_code == 1) {  // orange
    probability = 0.7;  // 70% prob of getting worst
} else if (color_code == 2) {  // blue
    probability = 0.3;  // 30%
} else if (color_code == 3) {  // green
    probability = 0.05;  // 5%
```

```
} else if (color_code == 4) {   // white
    probability = 0;
}

boolean result =  randomTrue(probability);
if(result == true){
      moreAssistance = 1;
} else {
      moreAssistance = 0;
}

trace("\nMora assistance? ");
trace(result);

return result;
```

2.1.4 Code Red – Emergency

The case in which the patient is a code red (therefore representing a high emergency situation) is treated differently: upon arrival at the reception desk, **he/she will have the highest priority and will be immediately moved to the correct department.**

If it is occupied, an attempt will be made to move the patient to the operating room used for patients with orange colour code.

For this reason, a function was created that controls the capacity of the room, in case there is a patient present, the procedure is as follows:

1. the patient is moved to the waiting room but always keeping the highest priority so as to allow him/her to be the first in the list;
2. the patient's colour is changed to allow him to go to the orange room.

Observation: the cases in which it was necessary to move the patient to another room were not managed, because as soon as either the red or orange room becomes free, the urgent patient is managed (this is a bit like what is managed in real cases, unless there is a cardiac arrest).

2.2 Assumptions about the scenario

To summarize what has been said above, the various assumptions made regarding the model are described:

- Each patient has a colour code assigned at check-in
- Patients with a red colour code receive immediate assistance without waiting
- Nurses are permanently assigned to rooms

Simplifications:

- Staff turnover has not been considered
- Procedure duration is modelled using probability distributions

2.3 WORKING FLOW'S IMPLEMENTATION

The system is structured as follows.

A patient can arrive and positions himself in the first node called **the startingNode** from there you follow a path until you get to a second node called **checkInNode**. In this node a queue of people will form who have to do the acceptance at the emergency room.

During the check-in, the nurse will give a colour code to the person; these can be
- **CODE** RED – EMERGENCY.
- **ORANGE CODE** – URGENCY.
- **BLUE CODE** – DEFERRABLE URGENCY.
- **GREEN CODE** – MINOR URGENCY.
- **WHITE CODE** – NO URGENCY.

Once the check in has been performed, we can have two cases
1. If the patient has a RED code, the agent follows the path named **EmergencyPath** which will take the patient directly to the **red room** where a team of doctors will visit/treat the patient (this team of doctors will always be available for emergencies and are not the nurses of the ED but directly of the hospital);
2. If the patient has a colour code other than RED, he will position himself in **the waiting room** (identified by the **waitingRoomNode node**).

Patients in the waiting room are eventually placed in a queue via a <u>***dynamic algorithm****</u> that assigns a priority.

If the **NursesRoomNode** is free, the first patient in the queue in the waiting room is called. Depending on the colour code, you can check if there is a free bed
- If there is a free bed then the patient is accompanied to the correct room
- If there is no bed, it remains in the NursesRoomNode until a bed is freed up.

2.3.1 Dynamic algorithm

One approach could involve hybrid management, which balances the severity of the case and the waiting time. This type of method is often called **dynamic priority management – dynamic priority based on severity and wait time.**

The idea is to assign each patient a <u>priority score</u> that changes over time, increasing as the waiting time increases. In this way, <u>even less severe patients will be treated sooner or later</u>, as their score will increase enough to surpass them on the list.

Scoring logic
<u>Color Code</u>: The initial severity of the patient determines a starting priority value:
- Code Red: Maximum priority (pt. 10000)
- Code Orange: High Priority (pt. 1000)
- Code Blue: Medium priority (100 points)
- Code Green: Medium-low priority (pt. 10)
- White Code: Background (pt. 1)

Wait time: Every minute of waiting increases the patient's priority score. The rate of increase depends on the color code:

- Orange: Rapid score increment (increment rate = 5).
- Light blue: Moderate score increment (increment rate = 3).
- Green: Moderate/slow score increase (rate of increase = 2).
- White: Slow score increment (increment rate = 1).

Scoring formula: Each patient will have a priority score calculated as

$$priorità = pt_{base} + incremento_t$$

pt_{base} is the score initially assigned based on the color code.
$incremento_t$ it is the score that increases over time, based on the minutes of waiting.

2.3.1.1 *Count Priority Formula*

```
double waitingTime = (now - p.arrival_time) / 60.0;

double basePriority = p.base_priority;
double timeIncrement = p.rate_increment * waitingTime;

return basePriority + timeIncrement;
```

In the queue: *agent 1 is preferred to agent 2 if*

```
countPriority(agent1, time()) > countPriority(agent2, time())
```

3 AGENTS

Different agents have been included in this project **.**

3.1 PATIENT

The emergency room of the San Donato Polyclinic in San Donato Milanese manages an average of between **65 and 80 patients per day**, with peaks in turnout during the holiday periods and when family doctors are not available, especially for seasonal ailments such as flu and Covid.

For this reason, when creating the *Patient* agent, it was chosen to use the value **70** as the starting value for the population.

Each patient has 5 priorities:

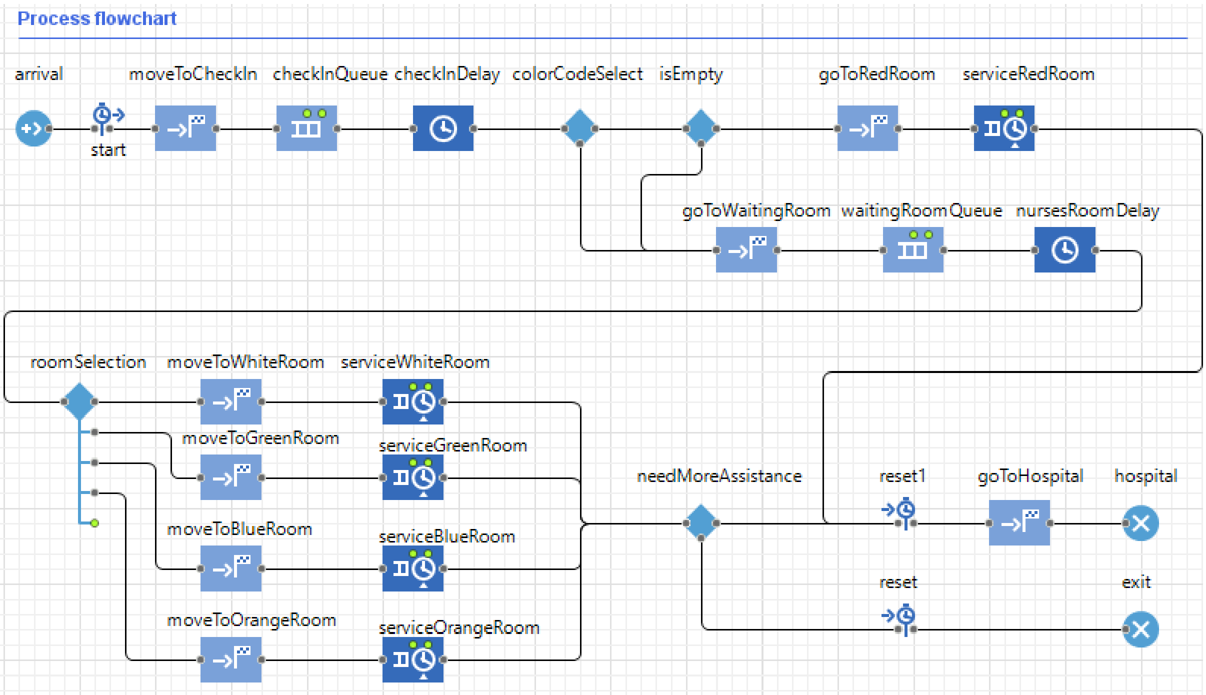
1. <u>Arrival time</u>, value expressed in **minutes** to represent the patient's arrival time
2. <u>Base priority,</u> which is given according to the colour code;
3. <u>Dynamic priority</u>, dynamically calculated priority to determine the waiting queue;
4. <u>Colour code</u>, a code that represents the patient's level of urgency. It is represented by integer values;
5. <u>Rate of increment</u>, a value used to calculate the patient's priority and is given in conjunction with the colour code.

3.2 NURSE

An additional agent is the nurse, who will take care of managing the patient. It does not have any property itself, but is managed as a resource.

4 FLOWCHART

Let's now investigate more about the components that make up the flow.



4.1 ARRIVAL

At the beginning of the process, a "source" called "*arrival*" was inserted, which represents the patients who arrive in the emergency room.
Since the arrival of patients is not precisely defined, it has been defined by an **exponential distribution**.
The Exponential distribution is frequently used to represent the time between random occurrences, such as the time between arrivals at a specific location.

When each patient arrives, he is assigned the arrival time:

```
agent.arrival_time = time();
```

4.2 CHECK IN

4.2.1 Queue

Patients in the queue for admission are placed in a queue following FIFO logic. The reason why the logic of colour codes is not followed is that in real hospitals, there must be an initial feedback from the nurse to observe if the urgency is real. After acceptance, in fact, in serious cases such as red codes, they skip the line and are immediately visited.

4.2.2 Delay

This delay is intended to simulate **admission** to the emergency room. As often happens, you have to wait for a certain time, which is why you represent the wait through a triangular variable time distribution **.**

Once the acceptance has been performed, the patient is assigned a colour code (randomly, but following certain probability values)

```
agent.codice_colore = assignColorCode();
```

AssignColorCode

```
double random = uniform(0, 1);
if (random <= 0.1) {
      person3D.setColor("Material__5__Surf", red);
      color_code = 0;
    // Red (probabiloty: 10%)
} else if (random <= 0.3) {
      person3D.setColor("Material__5__Surf", orange);
      color_code = 1;
      // Orange (probabiloty: 30%)
} else if (random <= 0.6) {
    person3D.setColor("Material__5__Surf", blue);
    color_code = 2;
    // Blue (probabiloty: 60%)
} else if (random <= 0.9) {
      person3D.setColor("Material__5__Surf", green);
      color_code = 3;
    // Green (probabiloty: 90%)
} else {
      person3D.setColor("Material__5__Surf", white);
      color_code = 4;
    // Red (probabiloty: 10%)
}

base_priority =(color_code == 0) ? 10000 : (color_code == 1) ? 1000 :
(color_code == 2) ? 100 : (color_code == 3) ? 10 : 1;
rate_increment = (color_code == 1) ? 5 : (color_code == 2) ? 3 :
(color_code == 3) ? 2 : 1;

trace("\nColor Code:");
trace(color_code);
```

The colour code is assigned randomly, but following a probability distribution. This is to make the scenario more real.

- **Code 0 (red):** 10% chance.
- **Code 1 (orange):** 20% chance.
- **Code 2 (blue):** 30% chance.
- **Code 3 (green):** 40% chance.
- **Code 4 (white):** If all conditions fail, the assigned code will be 4.

Depending on the colour code, the patient will have a series of parameters related to them, such as priority and increase rate.

4.3 Waiting room

After assigning the colour code, the patient (if it is not urgent) is placed in the waiting room.

To determine the maximum capacity, we considered some factors such as:
- The number of patients who can be treated in nursing rooms
- The average treatment time
- The Ability of Nurses to Manage Patients

Knowing that:
- The maximum number of patients per day is about 70;
- The check-in queue can hold about 100 people, but not all patients will remain in the waiting room, as those with a red code are handled immediately;
- The time spent in the waiting room depends on the waiting time to see a nurse and the average treatment time in the care rooms.

We performed the following calculations.

$$patient\ in\ 1\ h{:}\ 70$$
$$in\ a\ day = 24h$$

$$\frac{70}{24} = 3\ patients\ per\ hour$$

The calculation does not take into account that there will certainly be time slots with a certainly greater turnout. This is because we are only interested in identifying the capacity that the waiting room should have.
Assuming that a patient can stay a maximum of 3 hours in the waiting room, the maximum capacity could be equal to:

$$in\ 3\ h, the\ number\ of\ patients{:}$$
$$3 * 3 = 9$$

In the event that none of the patients was managed in 3 hours, but to take into account possible subsequent delays, we used a queue with a capacity of **30**.
This value was tested through different simulations, which allowed us to affirm the correctness of the value.

A constant that is not included in the simulation is the **Central Operations Agent**. This always knows the situation of the ED, so if a hospital were to stall, patients would be transferred to another hospital.

In this case, however, the queue is managed dynamically, as explained previously in the ***count priority formula*** section. Through a comparison of priorities, it is decided which patient is most urgent, in relation to the colour code and the time he has waited.

```
countPriority(agent1, time()) > countPriority(agent2, time())
```

4.3.1 Delay time

This can represent the time it takes for a nurse to call a patient for a visit, such as *uniform(1, 5)* to simulate random wait times.

4.4 PATIENT MANAGEMENT

Temporal distributions (triangular) were also used for patient management to simulate treatment time.

Waiting times differ according to the complexity of the operation and treatment, and according to the patient's colour code. This distribution allows you to maintain a balanced flow of patients based on severity, ensuring that resources are sufficient to avoid congestion and delays. If you notice that some rooms are overloaded in your model, you can adjust the number of nurses in each pool.

The **delay time** depends on the **severity** of the patients and the management capacity of the ward. The more severe the colour code, the longer and more complex the treatment could be. To determine a realistic treatment time, we can consider a few factors:

- **Patient severity**: Patients with higher urgencies require longer treatment times.
- **Treatment complexity**: More severe patients need more complex diagnoses and interventions.
- **Department capacity**: Rooms with more nurses or resources may be able to handle patients more quickly.

Every time the patient enters the room dedicated to him, the waiting time begins to be counted. In fact, this allows you to have time statistics about the single day

4.5 PROBLEM ENCOUNTERED: RED ROOM PREVIOUSLY OCCUPIED

In one of the simulations, this problem was observed: a patient with a red code arrived in the emergency room, but the room used for them was already occupied by a previous urgent patient. In order to manage these situations, it was decided to act in the following way (very much in line with the real scenario):

- If the red room is occupied, the patient is placed in the orange room;
- If the orange room is also occupied, the patient is placed in the blue room
- And so on.

In our case, given the waiting times and the capacities of the rooms, we only handled the case where the patient is routed to the orange room.

Below is the code used.

```
if(goToRedRoom.size() <= serviceRedRoom.queueCapacity){
      return true;
}
return false;
```

We also add that in order to place the patient in the orange room we have changed the color code to orange, but maintaining the priority of a red color code, thus allowing you to be the first to occupy the room given the emergency.

4.6 Closing Time

To obtain the statistics of a working day in the emergency room, a **closing time** *event* was used to end the process after 1440 minutes (i.e. after 24 hours). We know that the emergency room really never stops working (unless sporadic events are not treated in this scenario), but it is repeated that the stop of the flow has the purpose of analysing daily statistics.

5 Experiments and Simulations

5.1 Simulation Experiments and Results

In this section we report some experiments performed through the Anylogic simulator. Specifically, most of the considerations focus on the data categorized on October 9, 2024.

You can view the **python notebook** containing all the analyses in this file (statisticsProject.html). For each plot there are also comments related to the observed data.

5.2 KPI & KPS

In this section we report the key parameters of the system that have most influenced the behaviour of the model. The Key Performance Indicators instead better represent the overall results of the simulation.

5.2.1 Key System Parameters (KSP)

- **Room capacity**: The maximum number of patients each room can handle at the same time. This parameter directly affects waiting times and the ability of the emergency room to handle peaks in turnout.
- **Number of nurses**: The amount of human resources dedicated to each room, which determines how quickly patients are treated.
- **Distribution of arrivals**: The random distribution of patient arrivals greatly affects system overload. An exponential distribution can simulate the variability of arrivals throughout the day.
- **Priority Management Algorithm**: The rate at which patient priority increases over time, which balances patient severity with wait time.

5.2.2 Key Performance Indicators (KPI)

- **Average wait time**: The average time a patient spends waiting before receiving care. This KPI is crucial for evaluating the efficiency of the system.
- **Percentage of patients treated within an acceptable time**: How many patients are treated within a predefined time limit (e.g., less than an hour for code orange patients).
- **Resource utilization**: Percentage of nurse utilization in the various rooms. Excessive or underutilization can indicate inefficiencies in the system.
- **Number of untreated patients**: The number of patients who do not receive care by the closure of the emergency department.

5.3 ROOM EFFICIENCY AND WORKLOADS

From the results obtained by the model, it emerged that some rooms, in particular the **White Room** and the **Red Room**, suffer from frequent **overloads**, with waiting times often exceeding the defined critical threshold. This suggests that patients requiring more complex care (the most serious, such as those assigned to the Red Room) do not receive care quickly enough, with waiting peaks reaching worrying values.
In contrast, the **Green Room** and **Blue Room** rooms show a more balanced workload management, maintaining relatively low and stable waiting times. The **Orange Room** even seems underutilized, with very short waiting times and constant values, suggesting a possible **overallocation of resources** in this room compared to demand.

5.4 RESOURCE UTILIZATION

Analysis of resource utilization data indicates that there is an imbalance in resource allocation. In particular, the resources allocated to the White Room and Red Room are insufficient to handle the high demand, while resources such as those in the Orange Room could be better distributed in other areas to avoid overloads.
This indicates that a **reallocation of resources** could significantly improve the overall efficiency of the emergency department, reducing waiting times in critical areas and improving patient flow management.

5.5 KEY KPIS

During the project, several **Key Performance Indicators (KPIs)** were monitored, which provided key information on the functioning of the system:

- The **average wait time** highlighted areas with a high concentration of overload, particularly in the White Room and Red Room.
- The **percentage of patients treated within an acceptable time** showed that although most patients are treated on time, overloaded rooms such as the Red Room fail to fully meet the timelines for the most critical patients.
- **Resource usage** has confirmed that some rooms, such as the Orange Room, have resources that are not fully utilized, while other rooms are frequently above the overload threshold.

5.6 RECOMMENDATIONS

In light of the results obtained, it is possible to make some recommendations to improve the functioning of the emergency room:

- **Reduce overloads** in the most critical rooms, such as the White Room and Red Room, through a reallocation of resources or an increase in patient management capabilities in these areas.
- **Optimize resource allocation**: move part of the resources from the Orange Room, which appears underutilized, to rooms that show frequent overloads.
- **Constantly monitor KPIs**: Continue to monitor wait times and resource utilization to identify new opportunities for improvement and keep system efficiency high.

6 CONCLUSIONS

The present emergency room simulation project provided a detailed view of operational efficiency and resource management within a complex system. Through the modeling and analysis of different rooms and resources, we have obtained a series of results that allow us to better understand the internal dynamics of the emergency room and identify areas for improvement.

6.1 POSSIBLE FUTURE DEVELOPMENTS

To further improve the system, you could consider to:

- **Add more levels of severity** in patients to better differentiate critical cases from less urgent ones.
- **Implement a more sophisticated asset management model** based on artificial intelligence algorithms, which allow for more dynamic management of patient flow and staff allocation.
- **Simulate scenarios with variable patient flows**, such as emergencies or peak arrivals, to assess the resilience of the system under stressful conditions.

The simulation model demonstrated its usefulness in providing a detailed picture of the operation of an emergency department, identifying inefficiencies and suggesting potential improvements. Implementing the recommendations provided could lead to reduced waiting times, better use of resources, and ultimately increased efficiency of the healthcare system.

REFERENCES

Influenza accessi al Policlinico di San Donato. (n.d.). Retrieved from Il Cittadino: https://www.ilcittadino.it/stories/premium/sudmilano/influenza-covid-boom-accessi-al-policlinico-san-donato-o_107330_96/

PSD Carta Dei Servizio Febbraio 2023. (n.d.). Retrieved from Gruppo San onato: https://www.grupposandonato.it/mediaObject/ospedali/documents/Pol-San-Donato/trasparenza/PSD-Carta-Servizi-Febb2023/original/PSD+Carta+Servizi+Febb2023.pdf