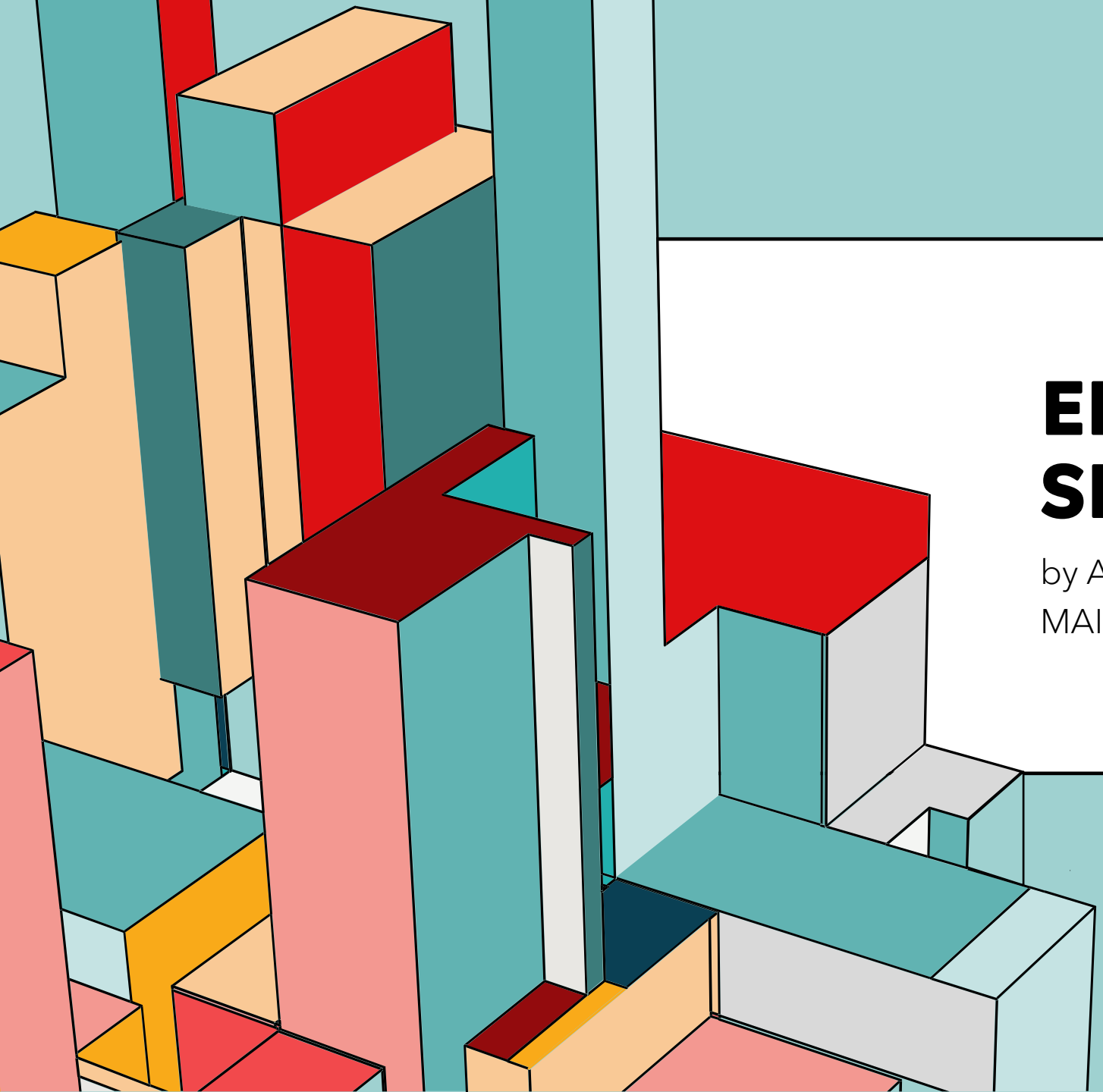
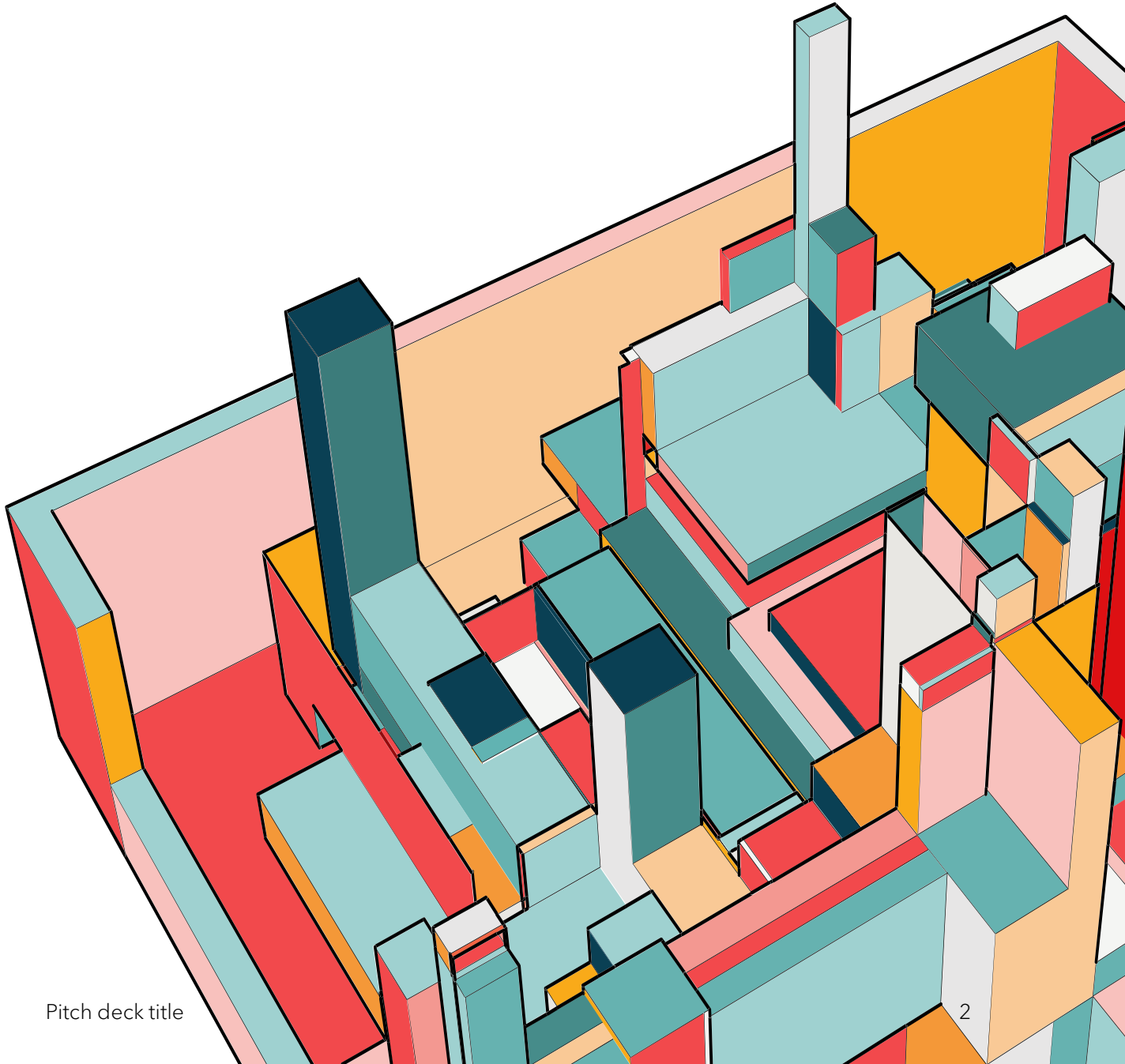


ELEVATOR SIMULATION

by Agustin, David and Yusuf
MAIA M1

ELEVATORS

Elevators allow you to travel through different levels of a building in an easier, more comfortable and faster way. The minimum elevator size is approximately 700 x 1,300 (W x D) millimeters versus a standard elevator size of 1,100 x 1,400 (W x D) millimeters. The elevator speed varies between 25 and 250 centimeters per second, although high performance devices can reach 70 kilometers per hour. This project consists in writing a program in python that will simulate the operation of an elevator that will serve at least 8 floors(level). The maximum number of passengers that the elevator can take is 8, with a total weight of 1600kg.



DESIGN

A Cabin

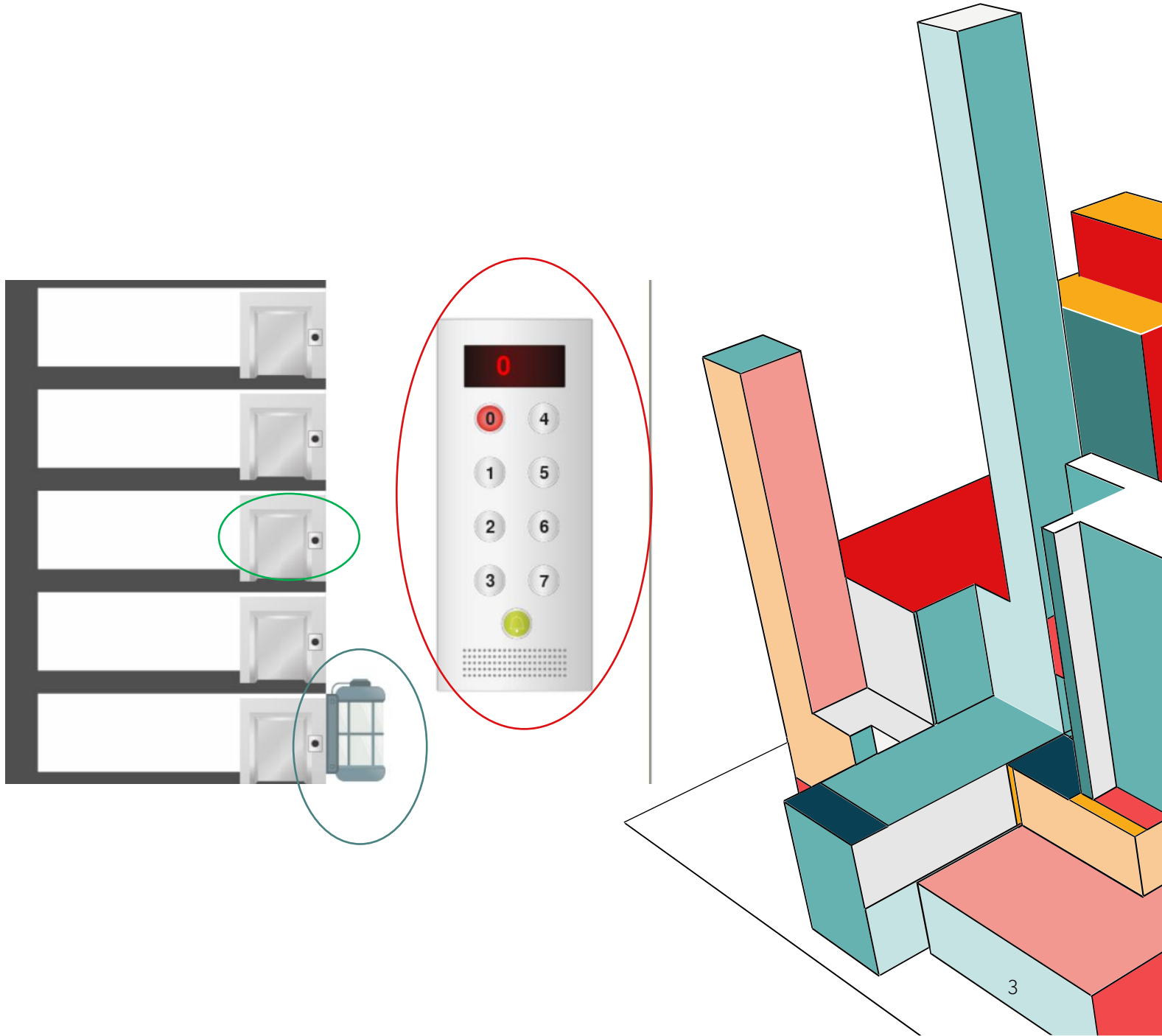
Represents the compartment that carries the passenger through the floors of a building when the elevator is moving.

Landing gates

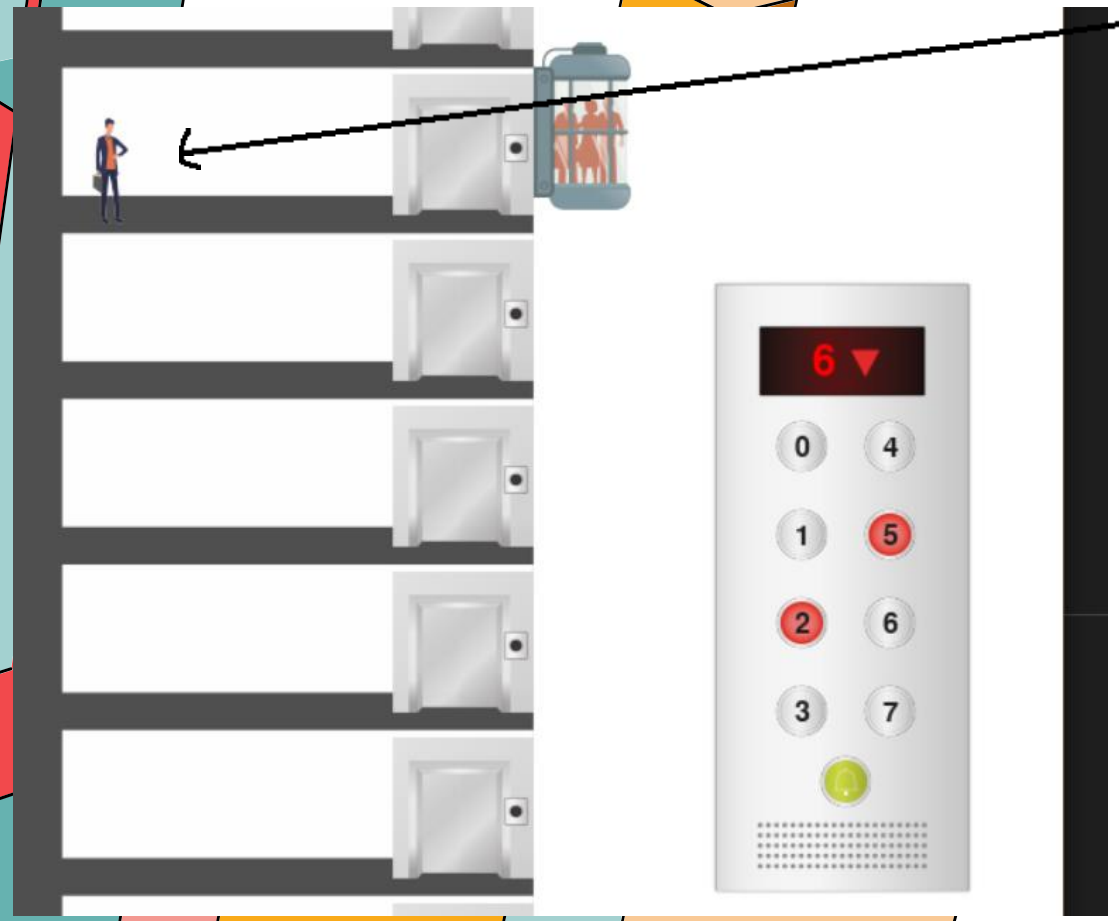
Landing gates: They are the device intended to isolate the car to better protect the passengers during the transport.

Control Panel

It allows you to call or direct the car. To do this, it has several buttons: call buttons, on each level of the building, and floor buttons, in the car itself.



FUNCTIONING

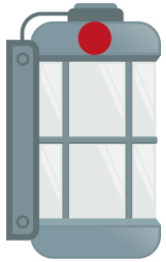


USER

- The user arrives in front of the elevator, moves forward and presses the button.
- The elevator opens and the user gets on the elevator
- The user presses the number of the floor he/she is going down to
- The elevator closes and performs the trip
- During this moment another user could ask for the elevator

FUNCTIONING: ELEVATOR

As long as it is not called anywhere, the elevator remains stationary

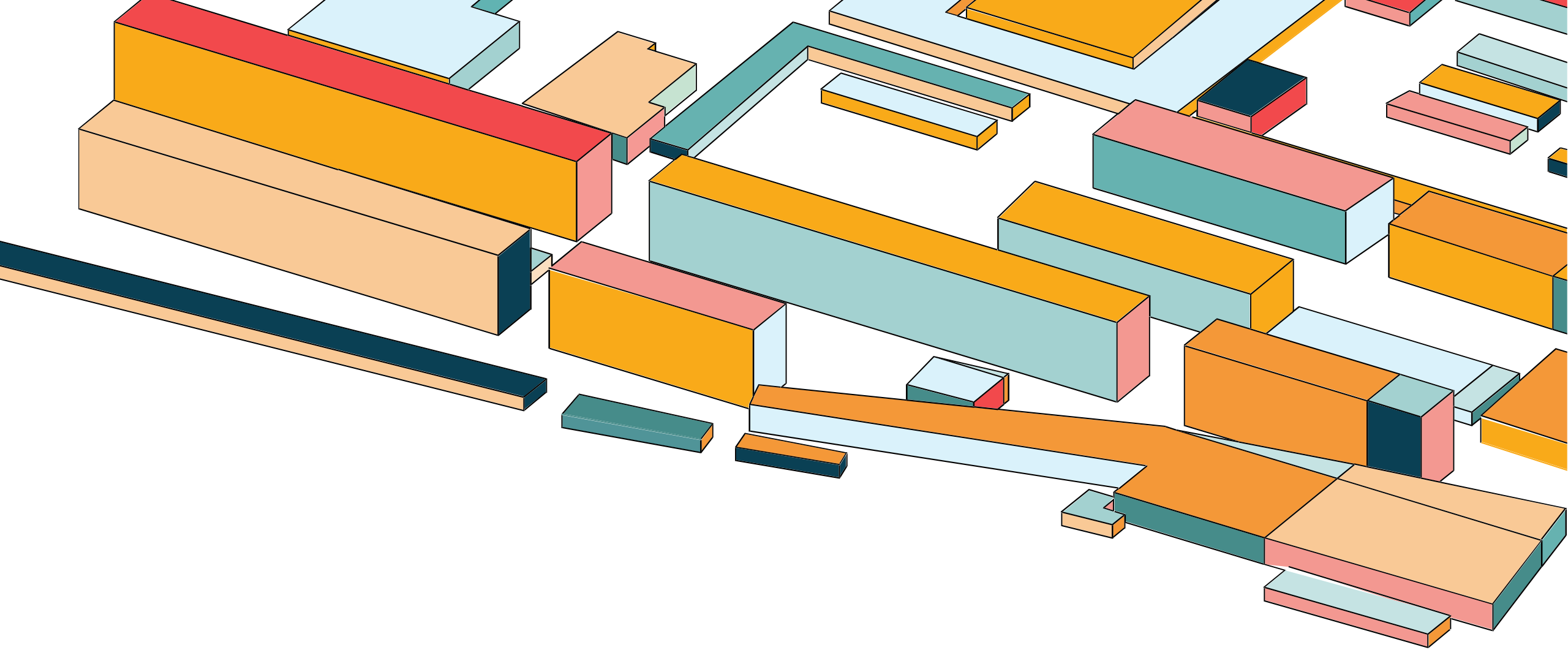


When called, Elevator starts to move to the floor which is requested.

While moving, it can be called to another floor be asked to go to a given floor.

It goes down only when it does not need to go up anymore. There are buttons to go up and down.

The movement of the elevator obeys a particular policy, to avoid the phenomenon of "starvation": as long as it has to go (called or sent) higher, it goes up (and respectively).



CODE OVERVIEW

CLASSES

Person

```
    def __init__(self, enter_floor, exit_floor,
weight=60, gender=True, age=32):
        """
        Initializes all the parameters and
assign in GUI the proper image asset according
to gender
        """
        self.weight = weight
        self.gender = gender
        self.age = age
        self.enter_floor = enter_floor
        self.exit_floor = exit_floor

        style = str(random.randint(1,3))
```

CLASSES

Building

```
class Building:
    IMG_NAME = 'building.png'
    def __init__(self, window):
        self.window = window
        self.img = ""

        self.load_image()
# Just asset configuration.
    def load_image(self):
        self.img =
pygame.image.load(os.path.join('src','img',
self.IMG_NAME))
        self.img =
pygame.transform.smoothscale(self.img,(300,
self.window.SCREEN_HEIGHT))

    def draw(self):
        self.window.screen.blit(self.img, (0,0))
```