# Der Aufzug



# Einleitung und finale Funktion

## Erste Ideen

Ich wollte ein Haus mit 3 Etagen, einem Fahrstuhl und zwei (oder mehr) NPCs, die den Fahrstuhl bedienen.

## finale Funktionen

Das Program ist in der Lage so viele Charactere gleichzeitig den Aufzug bedienen zu lassen wie man lustig ist.
Im Program selbst sind jedoch nur zwei charactere vorprogrammiert.
Der Aufzug priotisiert es die NPCs im Fahrstuhl an ihre gewünschte Position zu bringen und

sammelt erst neue Passanten ein, sobald die Charactere aus dem Fahrstuhl ihre Endposition erreicht haben.
Sollte sich bei der gewünschten Position eine andere Person befinden wird diese vom Fahrstuhl mitgenommen und ihre gewünschte Position wird zu der Liste hinzugefügt, die der fahrstuhl über Zeit abfährt.

## Link zu Scratch

---

link: https://scratch.mit.edu/projects/1013295896
Link zum Download:
https://github.com/GllejK/Informatik/blob/831553b8bae2b3b86cbcb4fced77b733fff4765b/Aufzugs%20Animation.sb3

# Prozess

## Schritt 1: Graphic Design

---

Ich habe mich nicht lange mit dem Design aufgehalten und habe nur ein "Haus" und einen Aufzug aus Rechtecken erstellt.
Die zwei NPCs habe ich aus den Vorerstellten Characteren genommen und nur umbenannt.

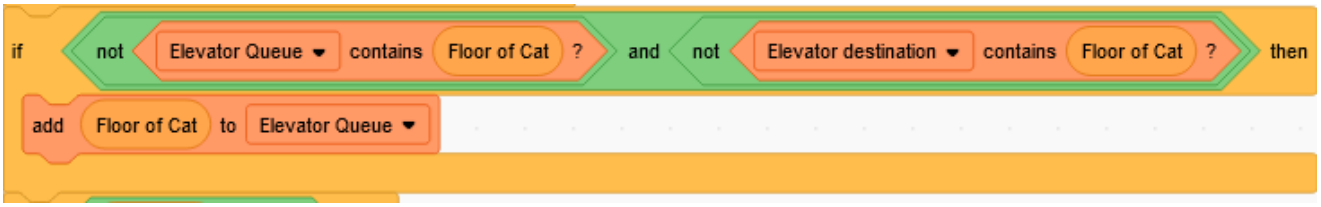
## Schritt 2: Der erste NPC - Cat

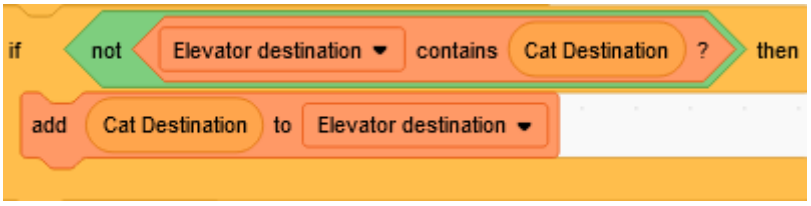
---

Zuerst habe ich für jede Etage Koordinaten ausgearbeitet, die als Ausgangsposition für "Cat" dienen. Zusätzlich habe ich eine Variable für die Position der Katze hinzugefügt.

Danach habe ich zu dem Skript diese Sequenz an Anweisungen hinzugefügt, welche eine zufällige Start- und Endposition auswählt. Dabei wird sichergestellt, dass es unmöglich ist, dass Start- und Endposition überlappen.
Auch habe ich die Liste "Floors" erstellt die am Anfang des Progammes auf ( EG , OG1 , OG2 ) gesetzt wird. Diese nutze ich, damit ich anstatt Zahlen Wörter für die einzelnen Etagen benutzen kann, was sehr bei der Übersicht und dem Debugging Prozess hilft.


Als nächstes habe ich die zwei Listen "Elevator Queue" und "Elevator Destination" hinzugefügt. Diese speichern die Reihenfolge, in welcher die NPCs abgeholt und weggebracht werden.
Ich werde ihre Funktion später noch genauer dokumentieren unter der Schritt 3: Der Aufzug - Elevator.

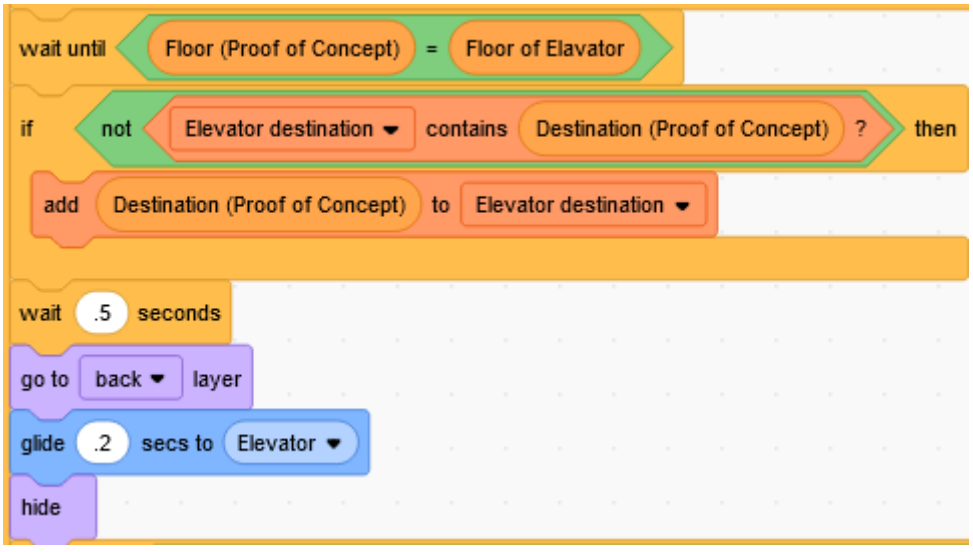



Danach habe ich mich um die animation für das verlassen des Aufzugs gekümmert. Ich habe für das umdrehen des Sprites Kostüme benutzt, weil es verhindert, dass das

Programm kaputt geht sollte man es mittendrin abbrechen.


Danach habe ich die Sequenz fertiggestellt, die das Einsteigen und Austeigen steuert.
Ich habe bei der Einsteigesequenz auch "go to back layer" genutzt um den NPC hinter dem Fahrstuhl zu verstecken und in danach transparent zu machen und erst bei der Aussteigeanimation wieder zeige. Dies erspart die Arbeit, die das Verfolgen von der Bewegung des fahrstuhls wäre.

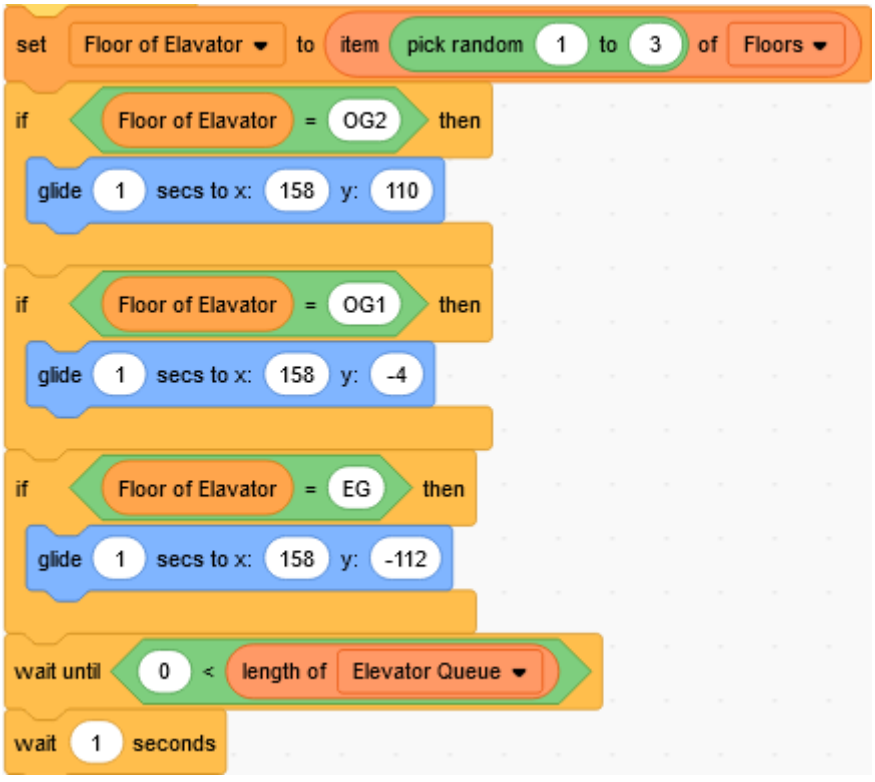



Am Ende habe ich die wichtige Reset Sequenz hinzugefügt, um Probleme bei einem Neustart zu vermeiden.


# Schritt 3: Der Aufzug - Elevator

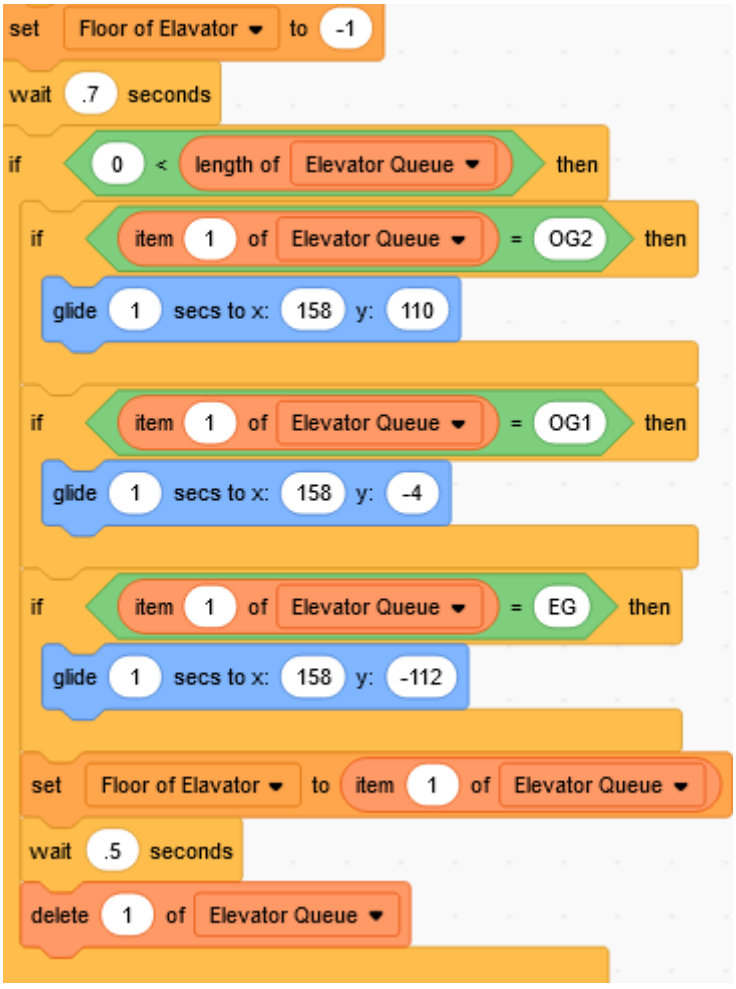
---

Zuerst habe ich eine Sequenz erstellt die eine zufällige Etage auswählt und sich auf diese Etage setzt.
Dabei habe ich eine Variable für die Position des Aufzugs erstellt.
Danach wartet der Aufzug darauf das eine Anfrage in die Warteschlange "Elevator Queue" geht.



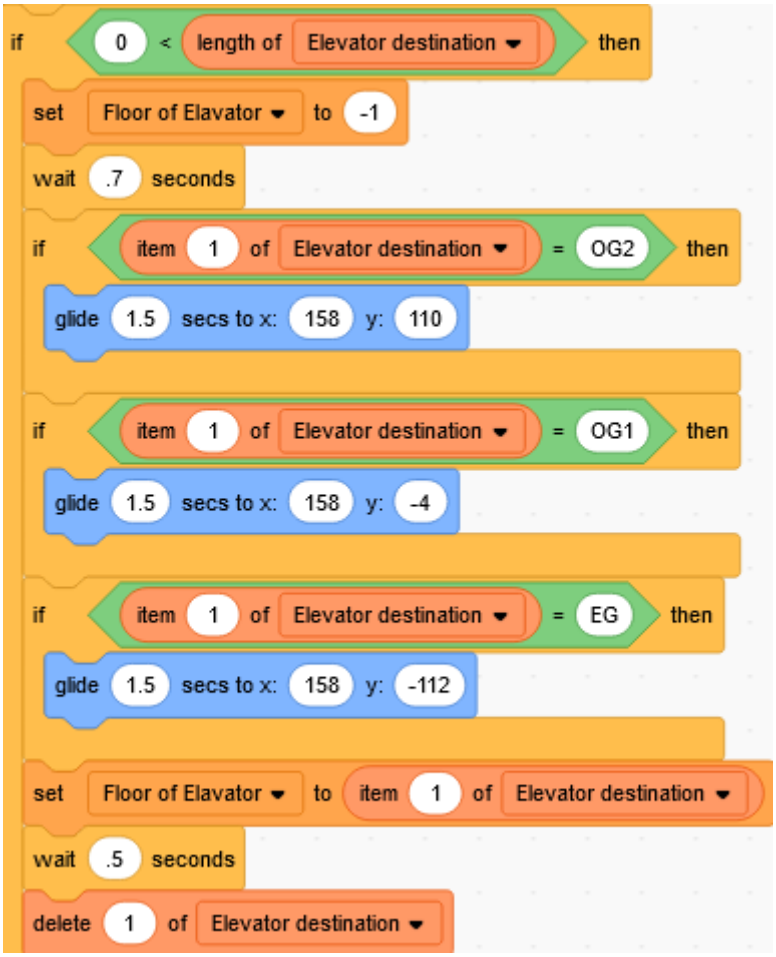

Hier ist die Sequenz für das Abholen von NPCs von der Warteschlange "Elevator Queue".
Dabei wird der Aufzug .7 Sekunden bevor er losfährt auf eine nicht existierende Etage gesetzt um zu verhindern, dass NPCs den Fahrstuhl betretten währen er sich bewegt.
Danach wird "Floor of Elevator" auf den obersten Eintrag in "Elevator Queue" gesetzt, sodass die NPCs ihre Einsteigesequenz anfangen können.
Danach wird ein halbe Sekunde gewartet, weil es sonst zu Problemen im Multithreading von Scratch führt (mehr dazu unter Probleme und Debugging) und der nun vergangene EIntrag

von "Elevator Queue" wird gelöscht.
Diese Sequenz ist in einer Falls, dann, sonst verzweigung gelagert, wobei die Sequenz nur angesprochen wird, wenn die Liste "Elevator Destination" leer ist.




Das hier ist die sequenz angesprochen wird sollten sich eine oder mehr Personen im Aufzug befinden, wodurch "len(Elevator destination)" >1 ist. Der Effekt ist quasi der gleiche wie der von der vorjerigen Sequenz. Diesmal werden jedoch die Einträge in "Elevator Destination" als Zielpunkte benutzt.
Dabei ist zu beachten, dass der Aufzug immernoch in der Lage ist NPCs aufzunehmen

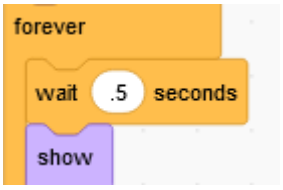
selbst wenn sie nicht direkt angesteuert werden.




## Schritt 4: Das Hinzufügen weiterer NPCs - Proof of Concept

---

Für das Hinzufügen eines weiteren NPCs muss man zwei neue Variablen erstellen, das skript von der Katze kopieren und die Variablen "Floor of Cat" und "Cat Destination" mit den zwei neuen Variablen ersetzt.
Dabei kann man die Wartezeit am anfang des Skriptes verlängern um Abwechselung in das Programm zu bringen




# Probleme und Debugging

## Scratch Multithreading

---

Das grösste Problem, mit dem ich zu kämpfen hatte, war das erzwungene und unvermeidbare pseudo Multithreading, dass Scratch anwendet.
Man kann es kaum beinflussen und es ist nahezu unmöglich irgendeine Art an Abfolge in das Programm zu bringen ohne die "Nachrichten" Funktio zu nutzen oder das Program ein ganzes Stück unhandlicher und langsamer zu machen.
Es hat mich um die 4 Stunden Trial-and-Error um das Timing zu lösen und es sollte nicht mehr zu diesen Problemen kommen.
Dieses Problem mit dem Timing des Skriptes war das einzigste Problem was ich hatte und trotzedem ein sehr verheerendes.