

Aufbau des Lagerorts

Die Klasse Lagerort wurde als Singleton implementiert. Dieses Objekt enthält eine Liste aller Gassen. Diese wiederum enthalten ein Regalbediengerät sowie ein linkes und rechtes Regal. Die einzelnen Regale beinhalten die Lagerfächer. Vom Lagerortobjekt aus kann also mit `GetInstance()` und den entsprechenden Get-Methoden des Objekts auf jedes einzelne Element des Lagerorts zugegriffen werden.

Simulation

Allgemein

Die Simulation wurde als Ereignisgesteuerte Echtzeitsimulation implementiert. Die Basis sind Lageraufträge wie Einlagerungs-, Auslagerungs- und Umlagerungsaufträge. Zu Beginn der Simulation muss (momentan) mindestens 1 Auftrag vorhanden sein, da die leere Auftragsliste, bzw. die daraus resultierende leere Ereignisliste, das Abbruchkriterium ist um die Simulation zu beenden. Dies könnte noch verbessert werden, indem die gewünschte Simulationsendzeit als Erinnerungsereignis in die Ereignisliste eingetragen würde. Die Simulation würde dann nach Abarbeitung dieses letzten Erinnerungsereignisses gestoppt, sofern mittlerweile keine noch späteren Ereignisse eingetragen wurden.

Es können während der laufenden Simulation weitere Aufträge hinzugefügt werden, die Startzeit des Auftrags muss allerdings die aktuelle Simulationszeit übersteigen.

Für jeden zukünftigen Auftrag wird ein Erinnerungsereignis in die Ereignisliste eingetragen mit Ereigniszeit = Startzeit des Auftrags. Dies dient dazu, dass die Simulation selbst nur die Ereignisliste abarbeiten kann, indem die Simulation jeweils das früheste Ereignis aus der Liste entfernt, anhand der Simulationszeit die benötigte Wartezeit berechnet und dann exakt zur richtigen Simulationszeit das Ereignis ausführt. In der Regel führt jedes Ereignis zu einem Nachfolgeereignis, welches wiederum in die Ereignisliste eingetragen wird.

Dieser Vorgang wird solange wiederholt, bis die Ereignisliste leer ist, wodurch die Simulation dadurch beendet wird.

Spezialfälle

Einzig das letzte Ereignis eines Auftrags sowie u.U. ein Erinnerungsereignis können dazu führen, dass kein Nachfolgeereignis mehr erstellt wird. Dies, wenn kein Auftrag im Allgemeinen bzw. für die aktuelle Gasse des letzten Ereignisses mehr besteht.

Würden keine Erinnerungsereignisse generiert, sondern direkt das erste Ereignis des entsprechenden Auftrags, wäre es schwierig, bei zusätzlichem Einfügen eines Auftrags, der vor dem anderen Ereignis startet, die Ereignisliste zu bereinigen. Dies wäre aber notwendig, da ansonsten zwei unterschiedliche Aufträge in derselben Gasse parallel ausgeführt würden. Dies ist aber physikalisch unmöglich und darf in der Simulation nicht geschehen! Bei jedem Erinnerungsereignis wird deshalb nur die Auftragsliste auf fällige Aufträge geprüft und bei Bedarf das erste Ereignis für diese Aufträge generiert. Es werden aber nur Ereignisse generiert für Aufträge von Gassen, welche nicht bereits durch andere, momentan ausgeführte Aufträge blockiert sind. Für diese Prüfung wird wiederum die Ereignisliste herangezogen.

Berechnung Simulationszeit

Die aktuelle Simulationszeit wird anhand folgender Formel berechnet:

$$\text{Aktuelle Simulationszeit [ms]} = \text{Simulationsstartzeit [ms]} + \text{Faktor} * (\text{Aktuelle Systemzeit [ms]} - \text{Startsystemzeit [ms]})$$

Da aber im Modus „As Fast As Possible“ nicht auf die exakte Ausführungszeit gewartet werden soll, sondern die Ereignisse sofort ausgeführt werden sollen, wird die Formel mit einem Korrekturwert ergänzt, welcher der Summe der Wartezeiten entspricht:








$$\text{Aktuelle Simulationszeit [ms]} = \text{Simulationsstartzeit [ms]} + \text{Faktor} * (\text{Aktuelle Systemzeit [ms]} - \text{Startsystemzeit [ms]}) + \text{Korrekturwert [ms]}$$

Der Korrekturwert bleibt bei der Echtzeitsimulation immer 0, weshalb für beide Simualtionsmodi dieselbe Formel verwendet werden kann.

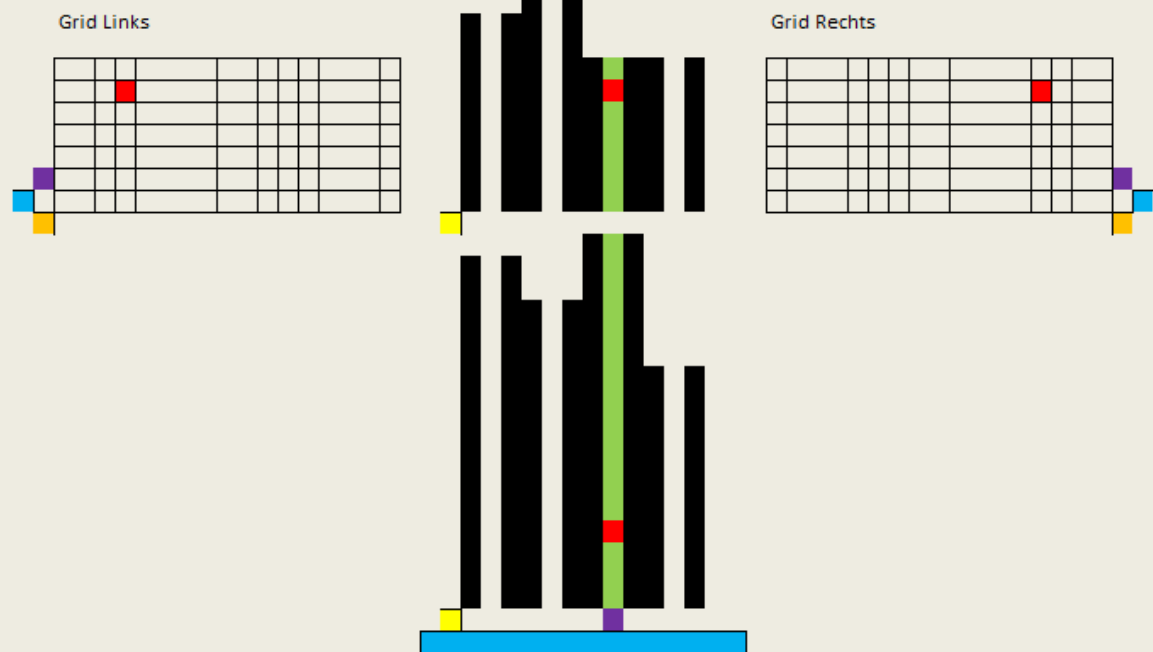
Fehler durch Rechenzeit und dessen Compensation

Es gibt nun aber Fälle, in denen die Ereignisse anhand der aktuellen Simulationszeit zu spät ausgeführt werden. Dies dadurch, dass mehrere Ereignisse die gleiche oder zumindest eine kurz aufeinanderfolgende Startzeit haben. Der Grund für die Verzögerung ist, dass ja das „ausführen“ der Ereignisse Rechenzeit benötigt und die Berechnung der aktuellen Simulationszeit auf der Systemzeit beruht. Diese Verzögerungen würden sich summieren und müssen unbedingt kompensiert werden. Also wird für die Nachfolgeereignisse nicht die aktuelle Simulationszeit als Basis zur Berechnung der Startzeit des Nachfolgeereignisses verwendet, sondern die Startzeit seines Vorgängerereignisses.

Die aktuelle Simulationszeit wird also nur verwendet, um die Wartezeit bis zur Ausführung des nächsten Ereignisses zu berechnen und um zu jedem Zeitpunkt der Simulation den entsprechenden Zustand bzw. die aktuellen Koordinaten der Regalbediengeräte zu berechnen.

-  Der Nullpunkt aller Koordinaten (X/Y/Z) genau in der Ecke
-  Der Nullpunkt des Lagergestells (Y/Z) genau in der Ecke
-  Der Laderoboter auf Ladeposition des bedeckten Lagerplatzes
-  Der Laderoboter auf Be-/Entladeposition
-  Das Lagergestell
-  Die Ein-/Ausgabestrasse
-  Die aktuell betrachtete Gasse

Aufriss



Grundriss