

## **Beschreibung Hauptprojekt**

## Rollfeld Simulation

Das Hauptprojekt des Praktikums ist eine Rollfeld Simulation (engl. Apron Traffic Simulation). Alle Flugzeugbewegung auf dem Rollfeld des Flughafens werden für einen Tag simuliert und protokolliert.

Es werden ausschließlich Flugzeuge am Boden simuliert.

Simulation werden bei Neu- und Ausbau von Flughäfen eingesetzt, um Effektivität teurer Baumaßnahmen zu beurteilen.

Nach Abschluss der Simulation müssen viel statistische übersichtlich präsentiert werden.

*Beispiel:* Für die Planung der Landbahn Nordwest am Flughafen Frankfurt, welche 2006–2011 gebaut wurde, war es eine entscheidende Frage, wie viel Lärm an welchen Stellen entstehen würde. Diese Frage wurde durch eine solche Simulation beantwortet, wie wir sie im Praktikum realisieren.

## Kernpunkte der Software

- ▶ Rollfeld Beschreibung im JSON Format einlesen; zoom- und scrollbare Darstellung.
- ▶ Beschreibungen von Flügen im JSON Format einlesen, Bewegungen der Flugzeuge grafisch darstellen; pausier- und vorspulbar.  
⇒ Simulation und Darstellung müssen unabhängig sein!
- ▶ Flugzeuge intelligent durch das Rollfeld zu leiten, ohne das es dabei zu Unfällen kommt. ⇒ Unfälle sicher erkennen!
- ▶ Statistische Auswertung durchführen, speichern und laden. Wie pünktlich wurden Flüge abgewickelt: Erwartungswert, Standardabweichung, Varianz und Extremwerte für Dauer und Länge des Rollwegs, Hotspots im Rollfeld, ...  
⇒ Eigenes JSON Format festlegen!

# Ein- und Ausgabe der Simulation

**Eingaben** Alle Eingaben sind Dateien im JSON Format

- ▶ Beschreibung des Rollfeldes als gerichteter Graph (geographische Punkte und gerichtete Kanten)
- ▶ Beschreibung von Flugzeugtypen
- ▶ Flüge: Wegpunktliste mit Uhrzeiten, welche von einem Flugzeug zügig abgearbeitet werden muss
- ▶ Verhaltensbäume zur Wegfindung der Flugzeuge
- ▶ Statistische Daten einer vorangegangenen Simulation

## Ausgabe

- ▶ Interaktive grafische Darstellung der Simulation; erlaubt zoomen, verschieben, pausieren, vorspulen
- ▶ Verständliche grafische Darstellung der Statistiken nach Ende der Simulation (z.B. Visualisierung von Hotspots und Erwartungswerten für Verspätungen)
- ▶ Ausgabe aller statistischen Daten im JSON Format

Vollständige Projektbeschreibung: per UniWorX „Blatt 04“

# Beschreibung Rollfeld: Knoten

Rollfeld besteht aus Knoten und Kanten.

- ▶ Jeder Knoten hat eine eindeutige Knotennummer und beliebig viele Knotennamen
- ▶ Ein **Knotenbezeichner** ist entweder eine Knotennummer oder ein Knotenname

*Beispiel:* Unter Name `runway1` können mehrere Knoten angesprochen werden.

- ▶ Position durch Koordinaten im WGS 84 Format gegeben
- ▶ Knoten können Liste von Knotenbezeichnern besitzen, welche bei Benutzung des Knotens frei sein müssen, inklusive aller daran angrenzenden Kanten

*Beispiel:* `[runway1, x]` alle Knoten mit Bezeichner `runway1` oder `x` müssen frei von anderen Flugzeugen sein

**Fallstrick:** WGS 84 Daten können die ganze Welt beschreiben; ein Flughafen ist im Vergleich zu Welt sehr klein.

⇒ Darstellung muss anfangs passen zoomen!

## Beschreibung Rollfeld: Kanten

Kanten entsprechen Rollwegstücken des Vorfeldes.

- ▶ Jede Kante ist angrenzenden zu genau zwei Knoten
- ▶ Flugzeuge bewegen sich immer geradlinig vom `in`-Knoten zum `out`-Knoten der Kante
- ▶ Kanten können eine maximale und minimale Geschwindigkeit haben
- ▶ Kanten können für manche Uhrzeiten und Flugzeugtypen gesperrt sein
- ▶ Flugzeuge drehen sich nur bei Erreichen eines neuen Knotens und der damit verbundenen Auswahl der nächsten Kante.

Der mögliche Drehwinkel ist abhängig von Flugzeugtyp und aktueller Geschwindigkeit!

## Beispiel: Flugzeugtypen

```
{ "airport": { "info": "Beispiel1: 2 Flüge mit 2 Flugzeugtypen."
, "inbound": 600, "outbound": 300
, "pushbacklag": 12, "pushbackspeed": 3.2
, "safetyground": 40.0, "saftetywaketurbulence":
  [{ "predecessor": "heavy" , "follower": "medium"
    , "distance": 5 }
  , { "predecessor": "medium", "follower": "medium"
    , "distance": 3 } ]
}
, "planes": [
  { "type": "Dash8-Q200", "wingspan": 25.9, "length": 22.25
  , "wakecat": "medium", "flightspeed": 120
  , "acceleration": 0.21, "safety": 3.75
  , "limits": [ { "speed": 50, "degree": 30 }
                , { "speed": 20, "degree": 90 } ] },
  { "type": "A350-900", "wingspan": 64.75, "length": 66.89
  , "wakecat": "heavy", "flightspeed": 140
  , "acceleration": 0.15 , "safety": 4
  , "limits": [ { "speed": 40, "degree": 30 }
                , { "speed": 10, "degree": 75 }
                , { "speed": 5 , "degree": 90 } ] } ] }
```

# Beschränkungen vs. Unfälle

**Beschränkungen** müssen jederzeit eingehalten werden!

Zum Beispiel darf ein Flugzeug niemals. . .

- ▶ ... eine gesperrte Kante befahren
- ▶ ... in größerem Winkel abbiegen, als Typ und Geschwindigkeit erlauben
- ▶ ... erlaubte Geschwindigkeit einer Kante über-/unterschreiten
- ▶ ... schneller beschleunigen oder bremsen als der Flugzeugtyp es erlaubt

**Unfälle** führen zum Abbruch der Simulation mit Hinweis

- ▶ Unterschreitung des geschwindigkeitsabhängigen Sicherheitsabstandes zwischen zwei Flugzeugen
- ▶ Verlassen des Rollwegs: keine weitere Kante erlaubt und kein rechtzeitiger Stopp möglich
- ▶ Knotenbeschränkungen verletzt werden

Erkennen von Unfällen muss demonstriert werden!



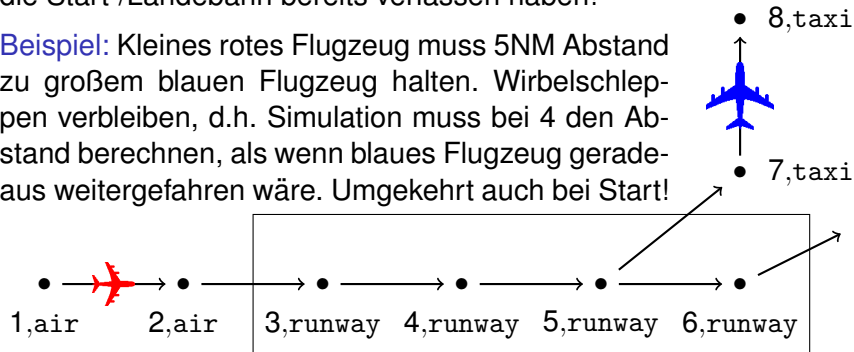
## Unfall: Wirbelschleppen

Flugzeuge hinterlassen gefährliche Luftverwirbelungen, welche besondere Typ-abhängige Abstände in der Luft benötigen.

Knoten-Bezeichner `runway` und `air` sind Knoten, bei denen diese besonderen Abstände gelten müssen!

Abstände müssen auch eingehalten werden, wenn Flugzeuge die Start-/Landebahn bereits verlassen haben!

**Beispiel:** Kleines rotes Flugzeug muss 5NM Abstand zu großem blauen Flugzeug halten. Wirbelschleppen verbleiben, d.h. Simulation muss bei 4 den Abstand berechnen, als wenn blaues Flugzeug geradeaus weitergefahren wäre. Umgekehrt auch bei Start!



## Beispiel: Flüge

Flüge beschreiben zu simulierende Bewegungen von Flugzeugen. Wegfindung ist Aufgabe der Simulation!

```
{ "flights":  
  [ { "aircraft": "A350-900", "schedule":  
      [ { "waypoint": "runway1", "time":21000}  
        , { "waypoint": "gate20", "duration":1200, "priority":1 }  
        , { "waypoint": "gate20", "time":24000, "duration":900}  
        , { "waypoint": "deicing", "duration":300}  
        , { "waypoint": "runway2", "time":24600, "priority": -3 }  
      ] }  
    , { "aircraft": "Dash8-Q300", "schedule":  
      [ { "waypoint": "runway1", "time": 21000}  
        , { "waypoint": "gatesBlissAir", "time": 21700  
            , "duration": 800 }  
        , { "waypoint": "runway1" }  
      ]  
      , "repetition":0.002  
    }  
  ]  
}
```

# Wegfindung

Flugzeuge müssen von der Simulation intelligent zu Ihrem nächsten Wegpunkt geführt werden, ohne dass dabei Unfälle passieren.

Intelligent erscheinende Wegführung der Flugzeuge wird durch Verhaltensbäume implementiert  $\Rightarrow$  eigenes Vorlesungskapitel

Simulation speichert Position und Geschwindigkeit von allen Flugzeugen.

Zusätzlich dazu jedes Flugzeug einen **inneren Zustand** welcher nur für den Verhaltensbaum relevant ist:

- ▶ Zielgeschwindigkeit
- ▶ Wegplan: Liste von Knoten-Bezeichner
- ▶ Alternativer Wegplan: Liste von Knoten-Bezeichner

Modelliert durch Klasse [Agent](#)

# Statistik

Informationen müssen ansprechend interaktiv vermittelt werden, d.h. Benutzer kann wählen zwischen Gesamtauswertung und Auswertung pro Flugzeugtyp oder Flugarten

- ▶ Anzahl simulierter Flüge
- ▶ Anzahl Start-/Landevorgänge
- ▶ Mittelwert und Standardabweichung von Verspätungen
- ▶ Mittelwert und Standardabw. gefahrene Strecke
- ▶ Hotspots der Orte (Knoten/Kanten) im Flughafen:
  - ▶ Wie war die Auslastung der einzelnen Orte?
  - ▶ Wo geschahen die meisten Verspätungen?
  - ▶ An welchen Orten mussten Flugzeuge unfreiwillig warten?
  - ▶ Welche Geschwindigkeit wurden auf welchen Rollwegstücken gefahren?

Statistik muss nach JSON exportiert und importiert werden können!

# Ablauf

- ▶ Neue git-Verzeichnisse für das Hauptprojekt anlegen  
Gruppenname muss wieder im Verzeichnisnamen vorkommen.
- ▶ Durch die Projektbeschreibung ( $\Rightarrow$  per UniWorX erhältlich) ist der Pflichtteil der Aufgabe vollständig gegeben. Es wird dazu noch kleinere Erweiterungen geben!
- ▶ Abgabe Code per UniWorX am 10.02.2019 um 23:59h.
- ▶ Endabnahme findet in der 7. Kalenderwoche 2019 statt, d.h. 11.-15.02.2019. Genauer Zeitpunkt wird mit dem Betreuer vereinbart.