

TicTacToeFlug (V1.0)

Flug erstellt am 10.01.2025

Geschätzte Flugdauer 15-30 min.

Schwierigkeitsgrad: Leicht

Aufgabe: Spiele Tic Tac Toe mit beliebigem Flugzeug.

Einleitung:

Lockheed Martin hat dem Flugsimulator ein „Tic-Tac-Toe Script Example“ beigelegt.

Dieses Example (Beispiel) ist

- grafisch nicht sehr ansprechend,
- man spielt gegen sich selber (kein Computergegner) und
- man steht beim Spielen schlicht auf der Startbahn.

Darum wollte ich herausfinden, ob man Tic Tac Toe auch

- in „schön“,
- mit einer „KI“ als Gegner und
- das ganze als Flug erstellen kann.

Dabei hatte ich einige Probleme zu lösen:

- 1) Ich kann nicht Scripten, also muss das Spiel ohne Scripts auskommen. Ist es möglich, nur mit Actions, Areas, Triggern und Objekten das Projekt zu verwirklichen?
- 2) Wie Erstelle ich das Spielfeld und die Spielsteine, ohne zusätzliche Szenario-Objekte zu erstellen, welche für die Mission extra installiert werden müssen?
- 3) Wie frage ich ab, welches Feld vom Spieler und welches von der KI belegt ist?
- 4) Wie frage ich ab, ob Spieler, oder KI gewonnen hat?
- 5) Wie sage ich der KI, dass sie am Zug ist?
- 6) Wie „Programmiere“ ich eine (wenn auch schlichte) KI, allein mit Actions, Areas, Triggern und Objekten?
- 7) Wie Programmiere ich eine intelligente KI? Es gibt viele verschiedene Spielsituationen, welche die KI erkennen / abfragen muss, um nicht zu verlieren.
- 8) Wie unterscheide ich nach dem Spielstart zwischen schlichter

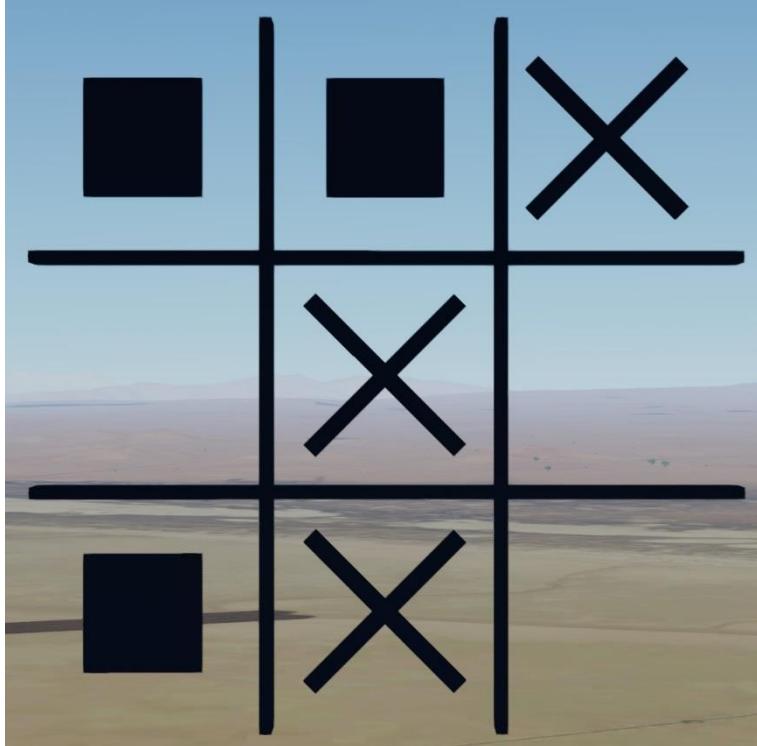
und intelligenter KI?

Wer Interesse hat, kann weiter unten meine Lösung der Probleme nachlesen.

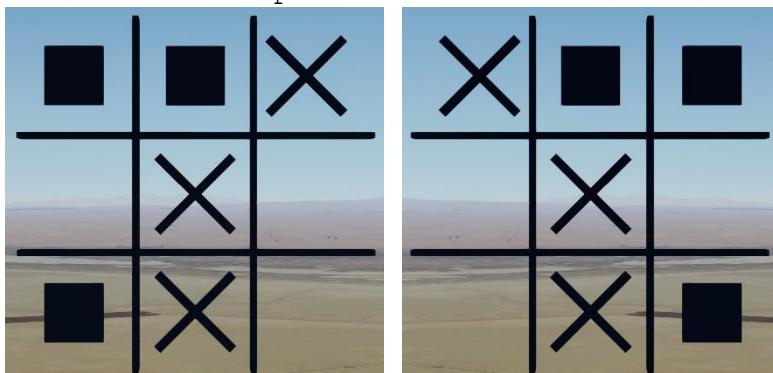
Das Spielfeld:

Wie Tic Tac Toe gespielt wird, sollte jedem bekannt sein.

Der Spieler hat als Spielstein ein X, die KI ein ■.



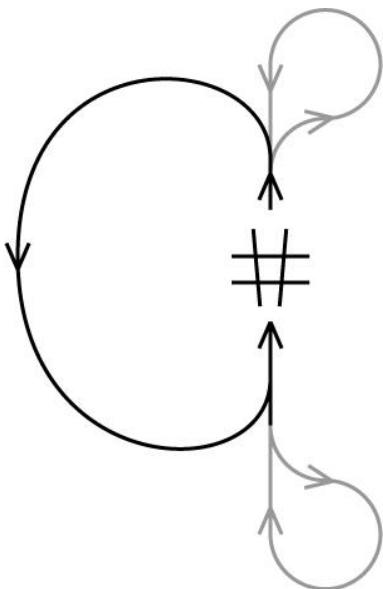
Bei diesem Flug - Tic Tac Toe gibt es eine Besonderheit:
Man kann das Spielfeld von zwei Seiten sehen.



Von der anderen Seite sind alle bereits gesetzten Spielsteine spiegelverkehrt.

Also ist nach dem Umdrehen ein „Umdenken“ erforderlich.

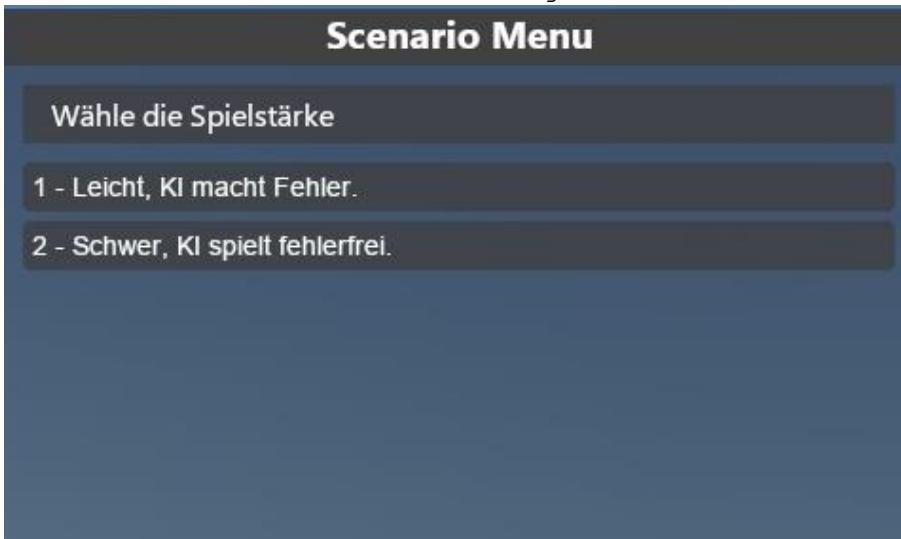
Du kannst von beiden Seiten durch das Spielfeld fliegen, hier zwei mögliche Flugrouten:



- 1) Eine Runde um das Spielfeld fliegen und immer von der gleichen Seite anfliegen.
- 2) Nach jedem Durchflug 180° drehen und das Spielfeld erneut anfliegen.

Der Spielstart:

- 1) Zu Beginn wählst du die Spielstärke der KI. Beim Modus „Leicht“ kannst du gewinnen, beim Modus „Schwer“ ist maximal ein Unentschieden möglich.



- 2) Anschliessend hast du 10 Sekunden Zeit, um das Flugzeug zu wechseln. Wähle ein beliebiges Flugzeug, auch Hubschrauber und Segelflugzeuge sind erlaubt (siehe weiter unten).
- 3) Das Spiel geht über 5 Runden, wer als erstes 3 Felder in einer Reihe, Spalte oder Diagonal belegt, gewinnt eine Runde. Eine Anzeige oben rechts zeigt den aktuellen Punktestand.



4) Berühre nicht die Spielfeldlinien.

Die Kollision schadet dem Flugzeug nicht, aber die Runde gilt bei Berührung als verloren.

Darum musst du mit grossen Flugzeugen sehr Präzise anfliegen. Eine Aussen Ansicht kann hilfreich sein.

5) Die erste Runde beginnt der Spieler. Starte und durchfliege ein beliebiges Feld für den ersten Zug.

Die Flugzeugwahl:

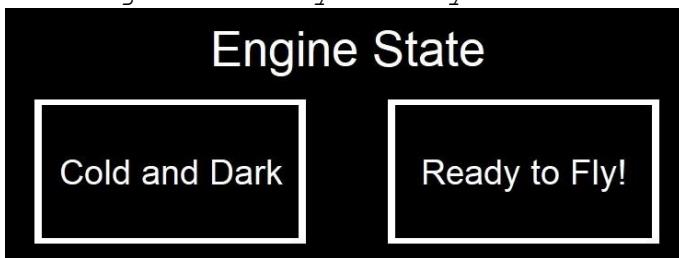
F16:

Falls du bei der F16 bleiben willst, musst du die Maschine startbereit machen.

Wähle dazu im Menü Vehicle/InstrumentPanel/EngineState:



Wähle jetzt Ready to Fly!



Segelflugzeug:

Bei Wahl eines Segelflugzeuges erkennt es das Spiel (am fehlenden Motor) automatisch und aktiviert Thermiken.

(Die Thermiken werden bei Segelflugzeugen mit Hilfsmotor nicht aktiviert, wähle also erst einen „reinen Segler“, nach Aktivieren der Thermiken kannst du auch zu einem Segler mit Hilfsmotor wechseln.)

Für den Start mit einem Segler musst du mit Shift+Ctrl+Y ein Schleppflugzeug anfordern.

Lasse dich auf eine geeignete Höhe schleppen und löse dich mit Shift+Y vom Schleppflugzeug.

Du musst keine Angst haben, dass dir die Aufwinde ausgehen:

- Der ganze Bereich um das Spielfeld ist mit einer Grundthermik belegt, welche bis 500 Fuss über Boden reicht.

- Vor und hinter dem Spielfeld befinden sich Thermiken, mit mittelstarkem Aufwind.

- Links und rechts vom Spielfeld befinden sich Thermiken, mit starkem Aufwind.

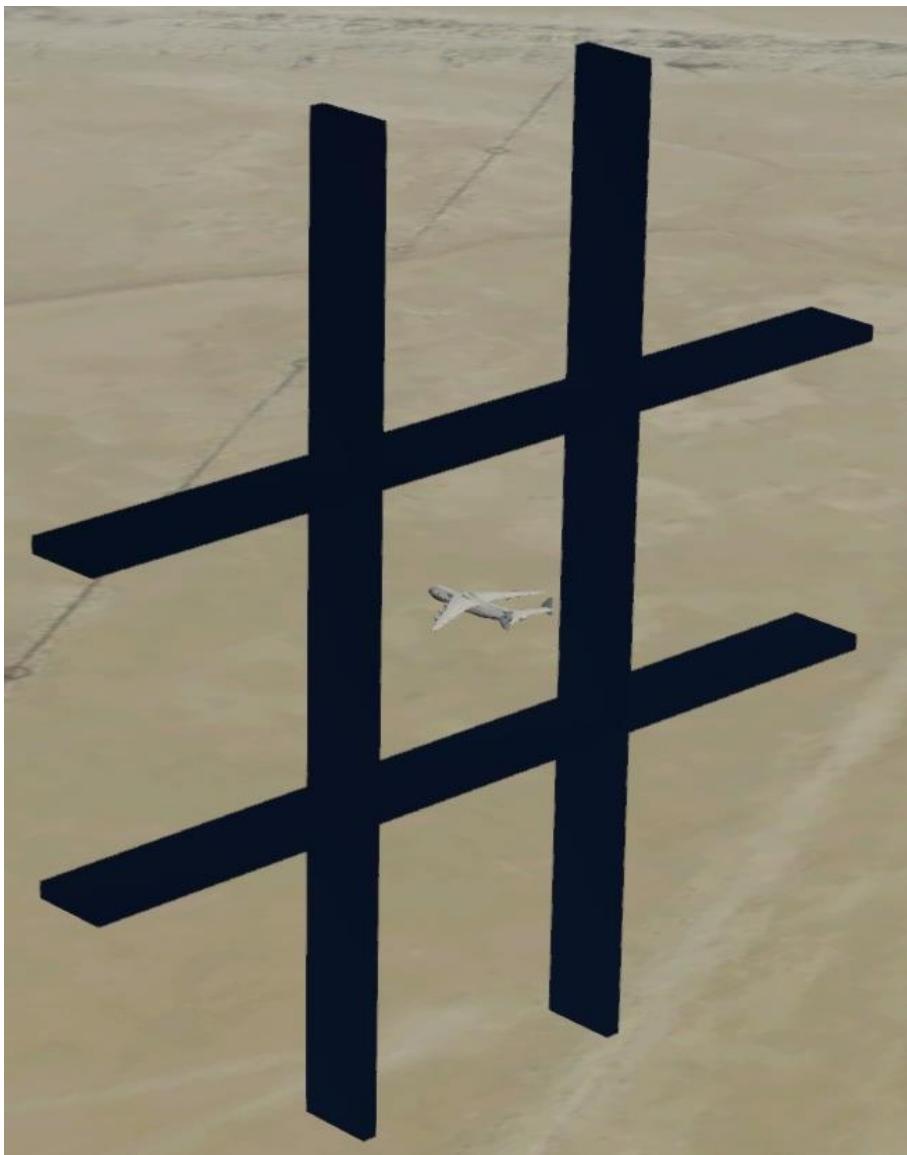
Zur Verdeutlichung hier mit grünen Rahmen dargestellt.



Die Thermiken wirst du an Vogelschwärmen erkennen.

Grosse Flugzeuge:

Das Spielfeld ist gross genug, dass du auch mit dem grössten Flugzeug, der AN225 Spielen kannst.

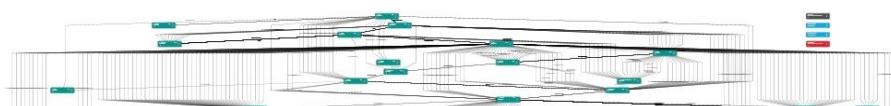


Wie ich das Spiel erstellt habe:

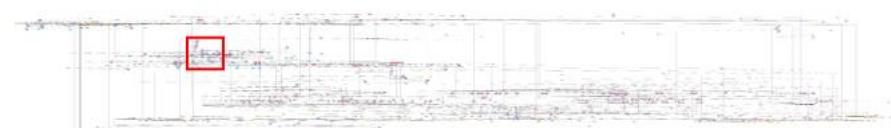
1) Ich kann nicht Scripten, also muss das Spiel ohne Scripts auskommen. Ist es möglich, nur mit Actions, Areas, Triggern und Objekten das Projekt zu verwirklichen?

Das ganze Spiel ist wirklich ohne Skripts erstellt. Nur 944 Objekte waren notwendig, ich habe schon Missionen mit 2000 Objekten erstellt. Allerdings sind beim Tic Tac Toe viel mehr Verknüpfungen zwischen den einzelnen Objekten vorhanden, als üblich.

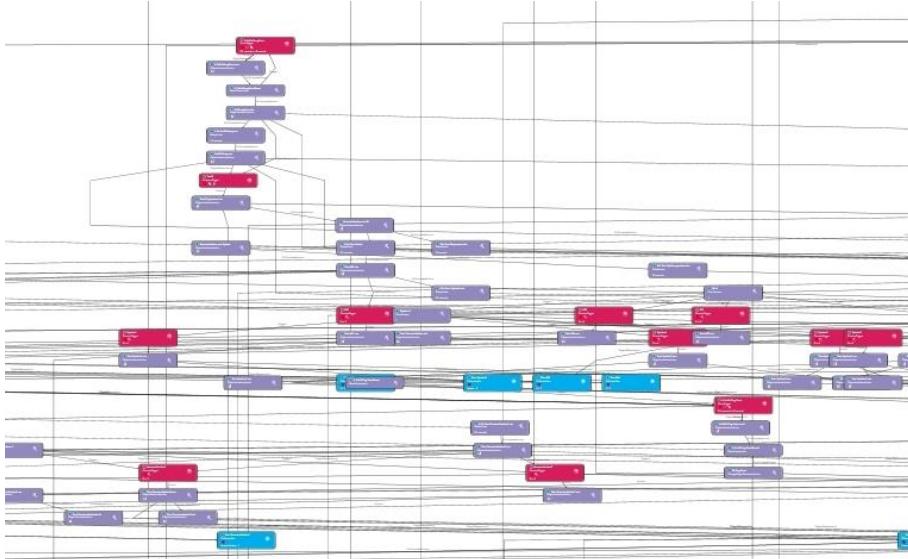
Gruppiert ist es noch einigermassen übersichtlich:



Ungruppiert sieht es allerdings übel aus:

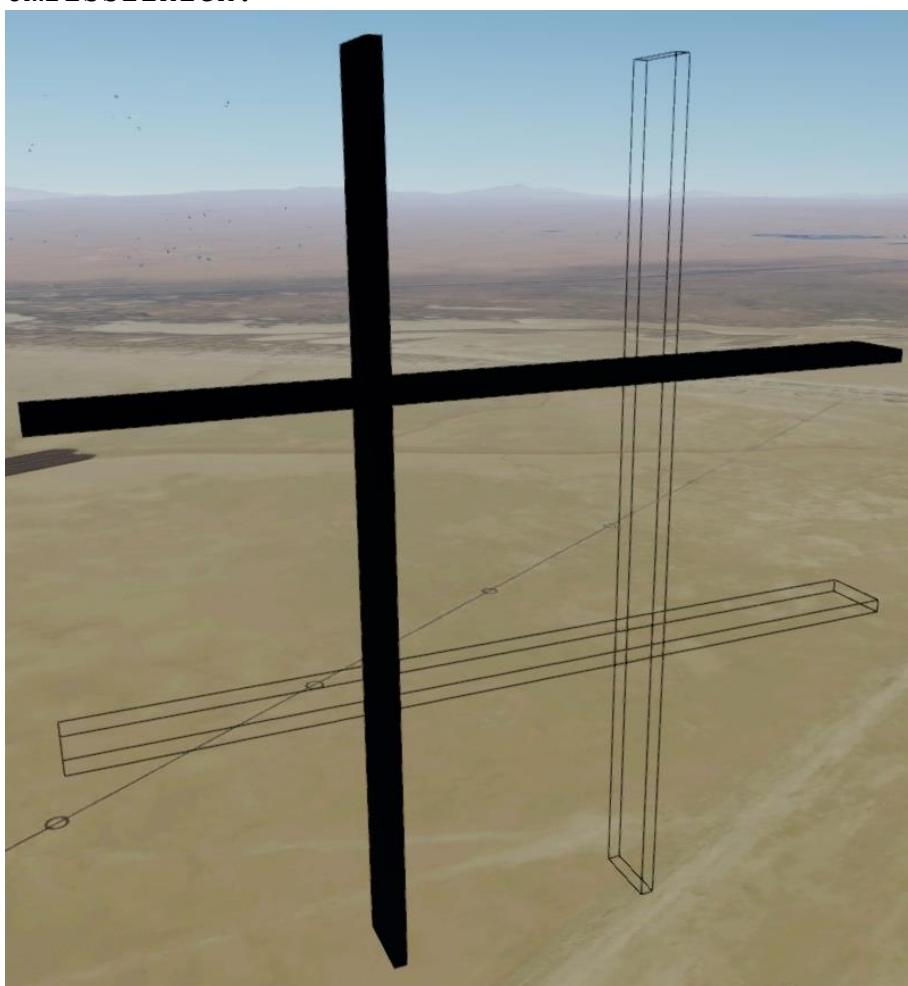


Detail (rotes Rechteck vom Bild oben):



2) Wie Erstelle ich das Spielfeld und die Spielsteine, ohne zusätzliche Szenario-Objekte zu erstellen, welche für die Mission extra installiert werden müssen?

Das Spielfeld habe ich schlicht mit farbig gefüllten Areas verwirklicht. Hier zur Veranschaulichung 2 Balken nur mit Umrisslinien.

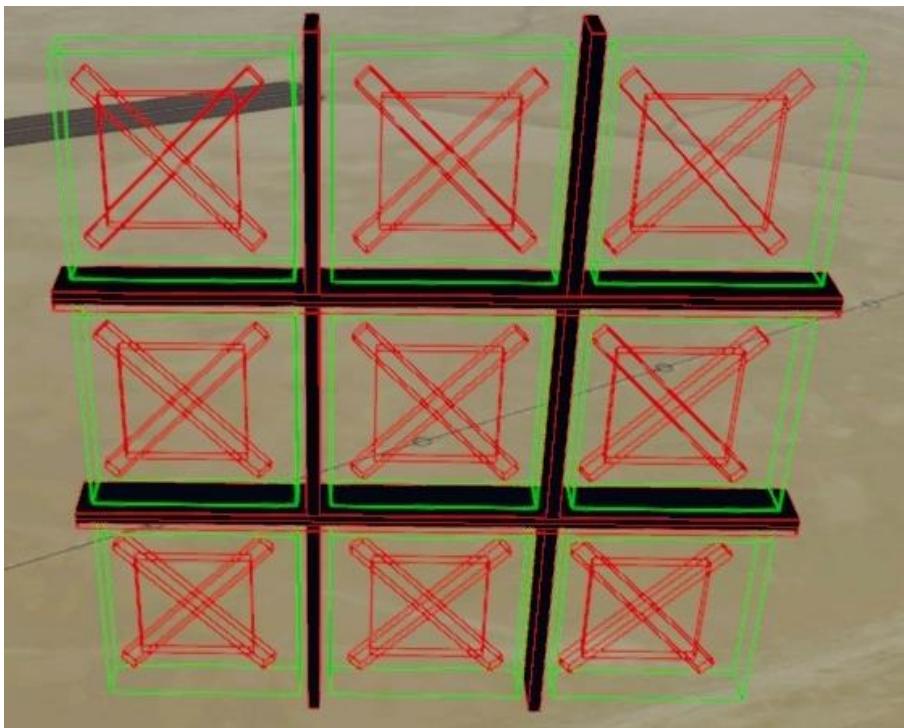


Die Spielsteine sind genauso erstellt.
Für ein X werden 2 Areas verwendet.

Damit der Spieler nachvollziehen kann, welchen Zug die KI gerade macht, wird das ■ „Blinkend“ eingeblendet (weiss, grau, schwarz).



Ein Problem seit P3dV5 ist, dass ich die Areas nicht „unsichtbar“ schalten kann, indem ich die Farbe auf Transparent stelle, es bleibt immer eine hauchdünne Umrisslinie sichtbar. Nur farblose Areas sind wirklich unsichtbar. Also habe ich im Spielfeld farblose Areas platziert (im Bild unten rot dargestellt). Mittels ChangeObjectPlacementAction wird bei Bedarf ein Spielstein, welchen ich im Boden versenkt (versteckt) habe an die Stelle der farblosen Area versetzt. Am Ende einer Runde werden die im Spielfeld platzierten Steine entfernt, also wieder im Boden versteckt.



Die grün dargestellten Areas sind übrigens Farblos und dienen dazu, mittels ProximityTrigger abzufragen, durch welches Feld der Spieler fliegt. Sobald der ProximityTrigger auslöst, wird die Area entfernt und durch eine andere Area ersetzt, welche dann beim Durchflug einen Text auslöst (Das Feld ist schon belegt). Weil die KI für einen Spielzug nicht durch ein Feld fliegt, muss beim Aktivieren eines ■ ebenfalls diese Area ausgetauscht werden. Klingt kompliziert? Ist es auch, jedenfalls wenn man das selber entwickeln muss.

3) Wie frage ich ab, welches Feld vom Spieler und welches von der KI belegt ist?

Bis jetzt weiss ich nur, welches Feld belegt ist, aber noch nicht, von wem.

Das war ein grosses Problem, denn mit PropertyTrigger kann ich sowas zwar abfragen, aber ich benötige 2x9 Parameter, welche ich

verändern und abfragen kann. Erst dachte ich an Flugzeug-Lichter, allerdings haben verschiedene Flugzeuge unterschiedliche (oder nicht alle) Lichter, welche ich benötige. Außerdem gibt es keine 18 Lichter, andere Parameter sind schwer zu finden und selbst wenn, könnte der Spieler gewinnen, indem er Panel, Strobe und Landing aktiviert.

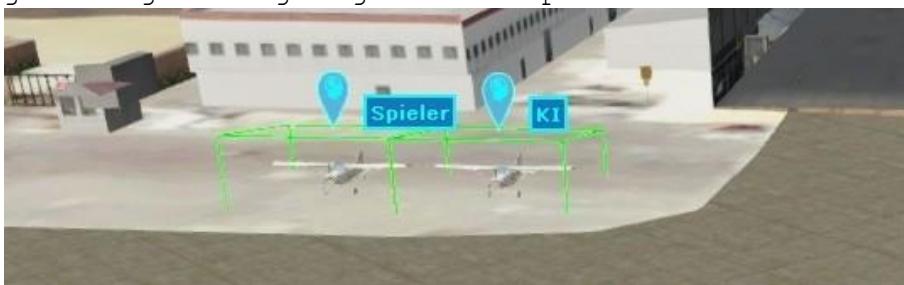
Endlich hatte ich die Lösung:

Man kann nicht nur die Lichter des Spieler-Flugzeuges manipulieren, sondern auch das von jedem KI-Flugzeug.

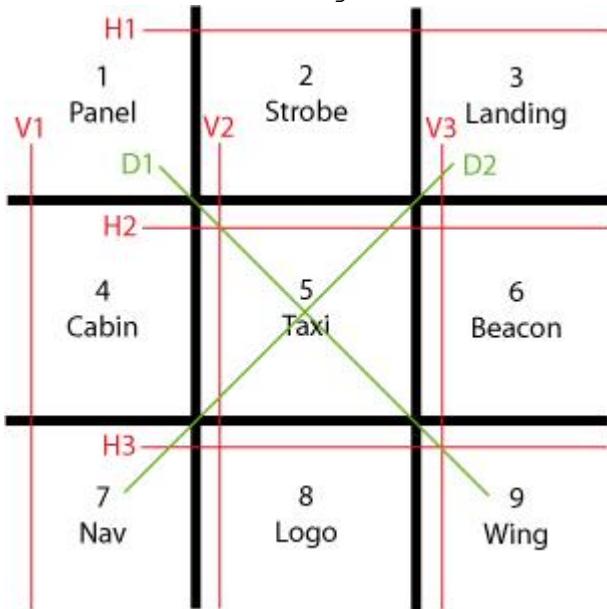
Also habe ich 2 KI-Flugzeuge, welche in allen P3d-Versionen (4/5/6) vorhanden sind auf dem Flughafen platziert.

Jedes Spielfeld bekommt bei beiden Flugzeugen das gleiche Licht zugeteilt.

Jetzt aktiviere ich bei Bedarf (mittels SetPropertyAction) beim jeweiligen Flugzeug das entsprechende Licht.

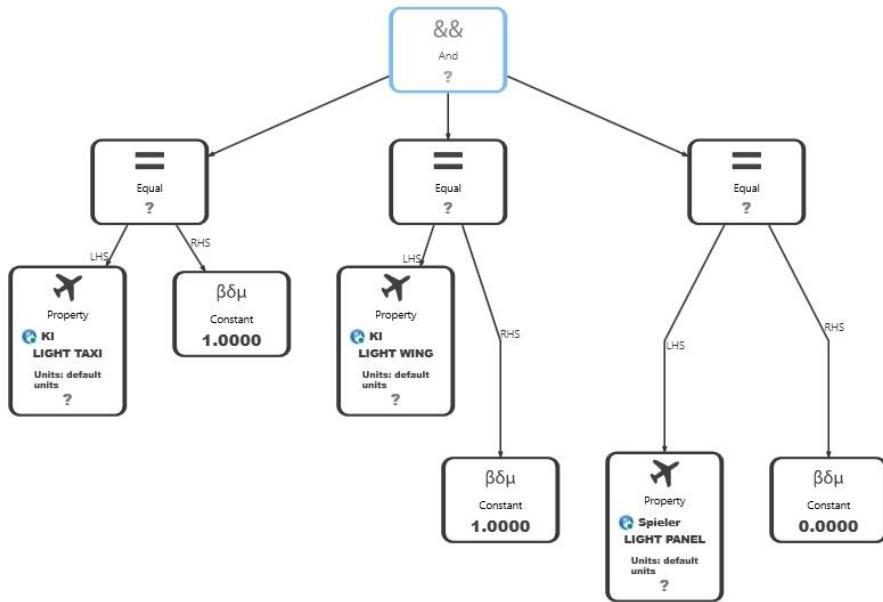


Hier die Zuteilung der Lichter:



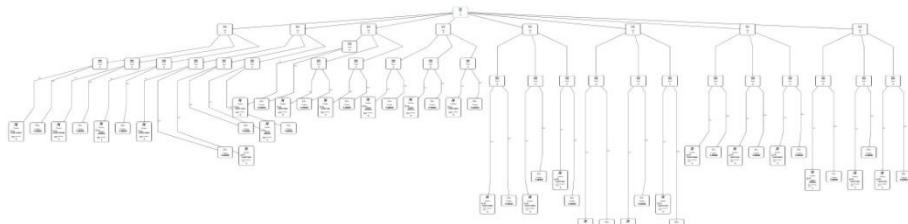
Mittels PropertyTrigger frage ich nun ab, welche Felder von wem belegt sind.

In diesem Beispiel fragt die Logik ab, ob in der Diagonalen (von oben links nach unten rechts), Feld 5 und 9 von der KI belegt und das Feld 1 nicht vom Spieler belegt ist. Wenn das der Fall ist, setzt die KI ihren Spielstein in das Feld 1 und gewinnt.

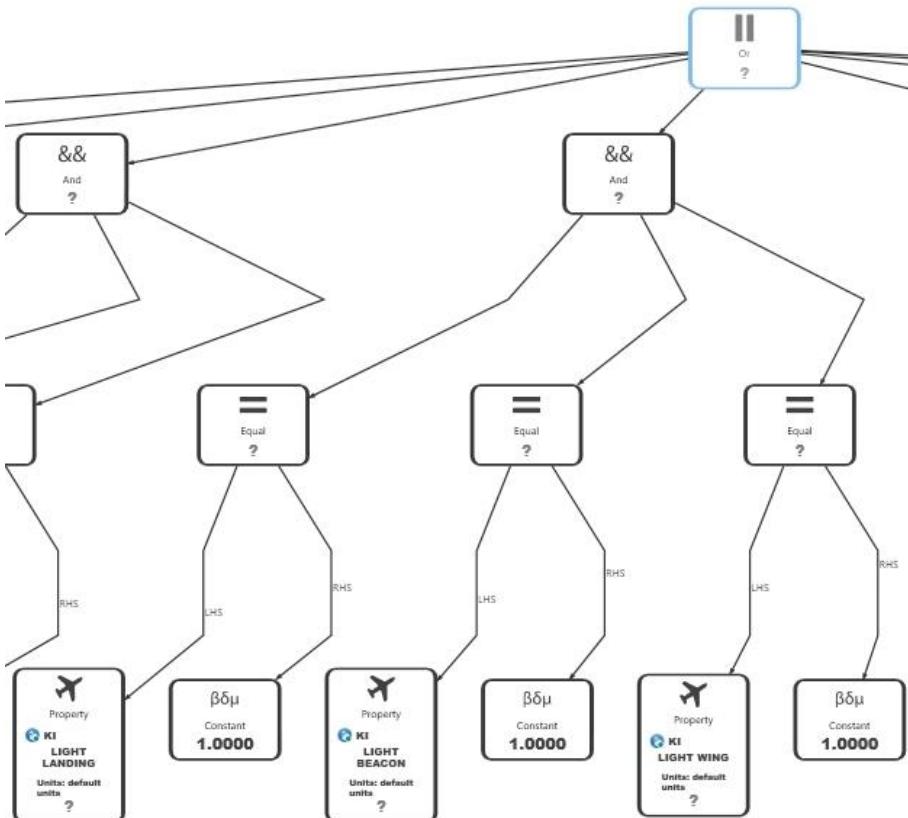


4) Wie frage ich ab, ob Spieler, oder KI gewonnen hat?

Wie in obigem Beispiel verwende ich auch wieder PropertyTrigger. Es werden nur 3 Abfragen benötigt (KI, Spieler, Unentschieden) allerdings ist die Logik bei den Abfragen umfangreicher, es werden bis zu 81 Logik-Elemente benötigt.



Es wird horizontal, vertikal und diagonal ausgewertet, ob die KI alle 3 Felder belegt.



Bei Unentschieden ist die Abfrage etwas anders, aber im Prinzip gleich:

Es wird abgefragt, ob alle 9 Felder belegt sind, egal, ob von KI, oder Spieler.

5) Wie sage ich der KI, dass sie am Zug ist?

Damit die KI „weiss“ wann sie am Zug ist, könnte ich doch schlicht nach jedem belegen eines Feldes durch den Spieler, die KI aktivieren.

Leider ist es nicht so einfach, denn es gibt Situationen, bei denen die KI keinen Zug machen soll, z.b. wenn der Spieler 3 in einer Reihe hat, also gewinnt.

Darum muss nach dem Spieler-Zug erst ausgewertet werden, ob der Spieler gewonnen hat, oder ob schon alle Felder belegt sind (= Unentschieden).

Darum aktiviert ein Spieler-Zug die Spielfeld-Auswertung und eine ProximityAction (namens KI-Zug), welche beim Verlassen (OnExit) der zugehörigen Area „KI-Zug“ den KI-Zug auslöst.

Nebenbei:

Warum so kompliziert, man könnte den KI-Zug doch nach Abschluss der Spielfeld-Auswertung starten?

Das funktioniert nicht, weil die Spielfeld-Auswertung auch nach jedem KI-Zug gestartet wird. Es würde also laufend ein weiterer KI-Zug ausgelöst, der Spieler käme nie zum Zug.

Ok, wenn du meinst, man könnte ja 2 Spielfeld-Auswertungen machen, eine für Spieler und eine für KI, hast du recht, allerdings werden für die Auswertung 183 Objekte benötigt.

Wie gesagt, Spielfeld-Auswertung und ProximityAction starten, sobald die Spielfeld-Auswertung weder „Spieler hat gewonnen“, noch „Unentschieden“ ergeben hat, löst sie das Entfernen der KI-Zug-

Area aus.

Das Entfernen der Area funktioniert in gleicher Weise wie das Setzen der Spielsteine, die Area wird schlicht an einen anderen Ort versetzt.

Weil die ProximityAction auf verlassen (On Exit) reagiert, ist ein verschieben der Area an eine andere Stelle das gleiche, wie ein Verlassen der Area.

Ergibt die Auswertung, dass der Spieler gewonnen, oder keine freien Felder vorhanden sind, wird die ProximityAction (KI-Zug) deaktiviert.

Ein Deaktivieren bewirkt, dass beim Verlassen der zugehörigen Area kein KI-Zug ausgelöst wird.

Die Area ist übrigens so gross, dass der Spieler diese nicht verlassen wird, selbst wenn er mit Kampfjet und voll Speed fliegt. Anfangs hatte ich die Area so geplant, dass sie nach 0.5 Sekunden Flug verlassen wird, das brachte aber weitere Probleme mit sich, z.b. wenn man mit Hubschrauber fliegt und sich nicht weit vom Spielfeld entfernt, oder mit Kampfjet bei voll Speed aus der Area fliegt, bevor die Auswertung beendet ist.

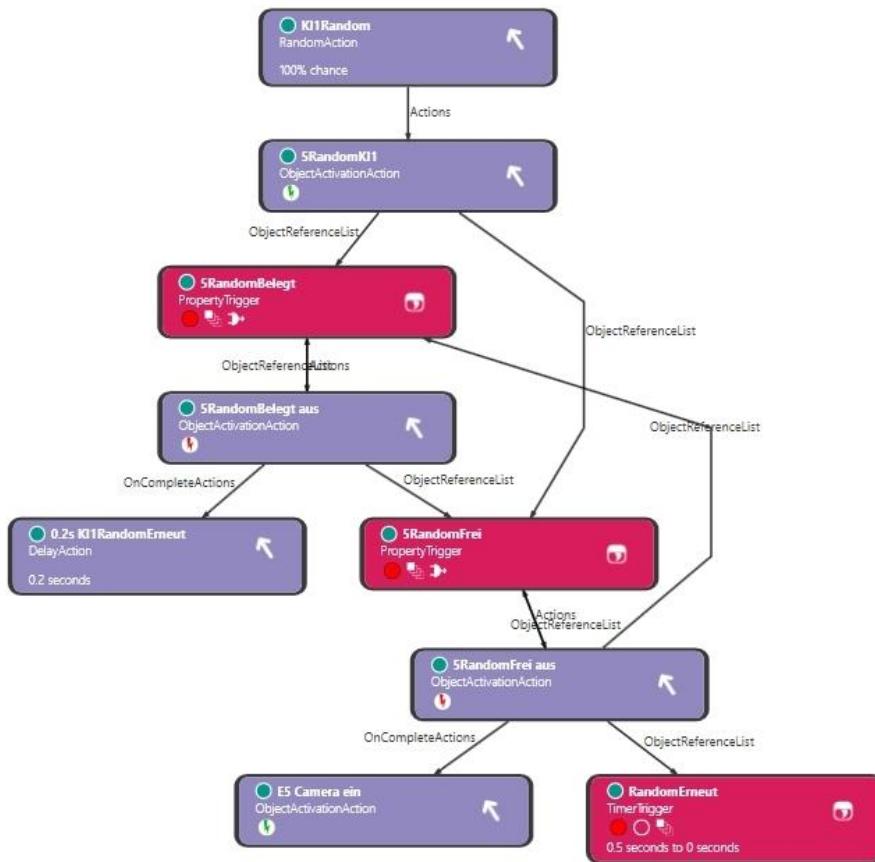
6) Wie „Programmiere“ ich eine (wenn auch schlichte) KI, allein mit Actions, Areas, Triggern und Objekten?

Sobald die KI am Zug ist, wird eine RandomAction ausgelöst.

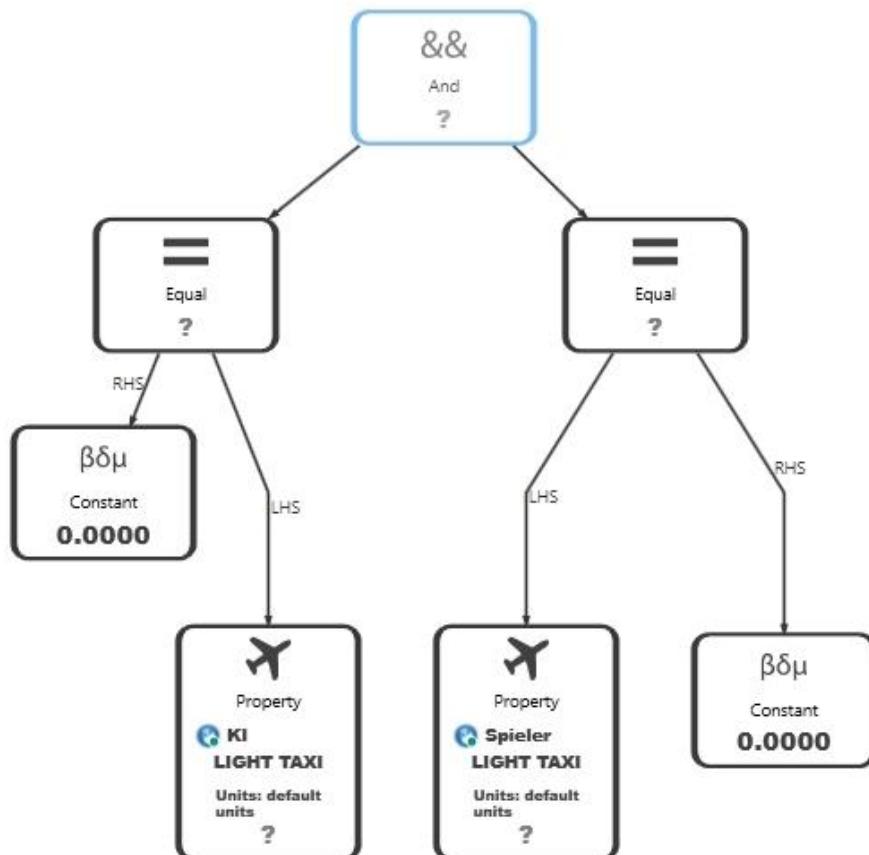
Diese Action wählt zufällig eines der 9 Felder aus und aktiviert zwei PropertyTrigger (Frei/Belegt).

Bei „Frei“ belegt die KI das Feld, bei „Belegt“ wird die RandomAction erneut ausgelöst.

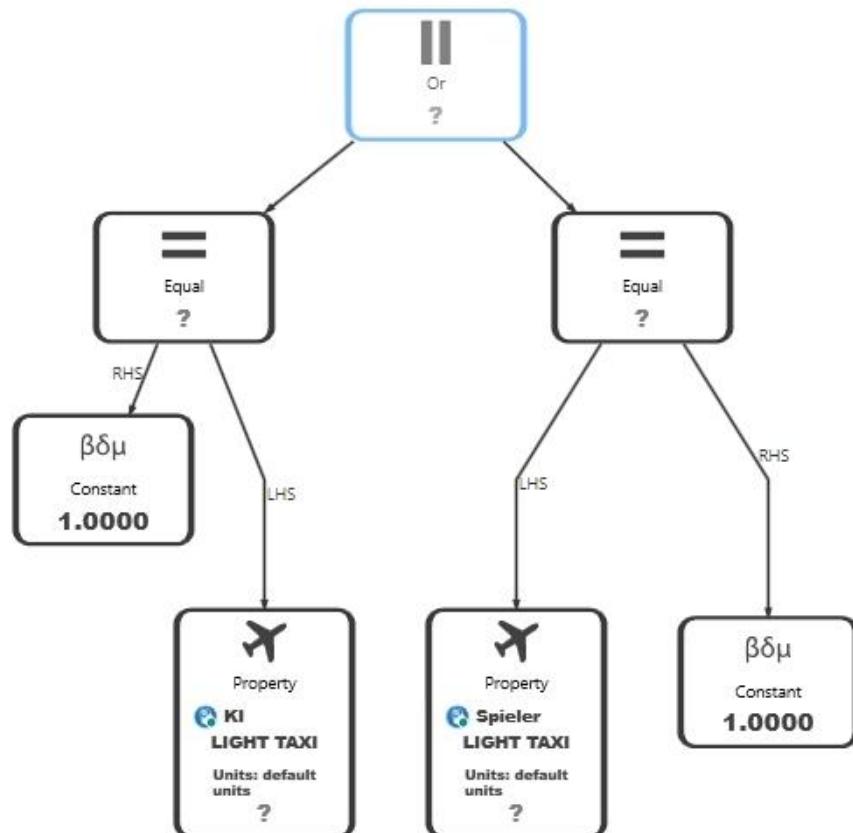
Aus unerfindlichen Gründen hat das nur in 95% der Fälle funktioniert, darum habe ich mit 0.5s Verzögerung die RandomAction erneut auslösen lassen. Sollte das Feld frei sein, wird diese verzögerte Action sofort deaktiviert.



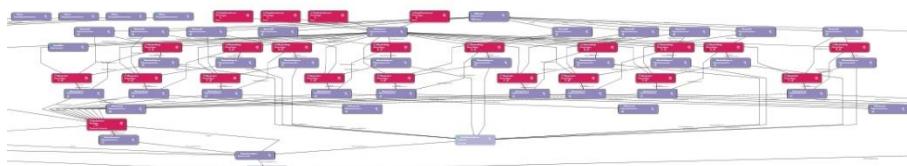
Die Logik von „Frei“ ist recht simpel:
 Es wird abgefragt, ob Light Taxi bei KI und Spieler Flugzeug ausgeschaltet ist (Constant=0).



Die Logik von „Belegt“ ist entsprechend:
Es wird abgefragt, ob Light Taxi beim KI oder Spieler Flugzeug eingeschaltet ist (Constant=1).



Einzelne sieht das übersichtlich aus, aber macht man das mit allen 9 Feldern und verknüpft sie entsprechend, werden 65 Objekte benötigt:



KI-Leicht wählt zufällig ein freies Feld, wie mache ich die KI intelligent?

7) Wie Programmiere ich eine Intelligente KI? Es gibt viele verschiedene Spielsituationen, welche die KI erkennen / abfragen muss, um nicht zu verlieren.

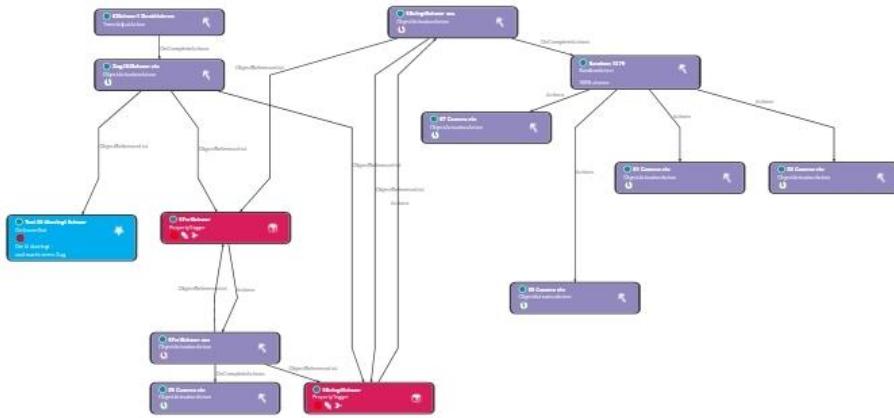
Die intelligente KI war eine grosse Herausforderung.

Vor allem Zug 2 war schwer zu verwirklichen.

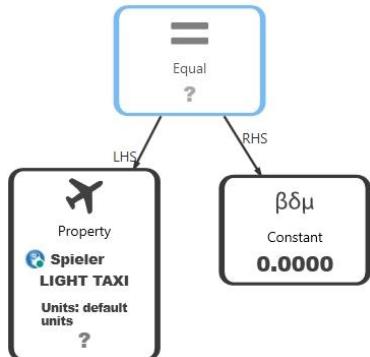
Denn genau mit Zug 2 entscheidet sich, ob die KI gewinnen kann.

Doch der Reihe nach:

1. Zug KI: Hier wird schlicht abgefragt, ob Feld 5 (das mittlere) belegt ist, wenn nein, wird das Feld von der KI belegt, wenn belegt wird zufällig Feld 1, 3, 7, oder 9 belegt.



Die zugehörige Logik ist entsprechend einfach:

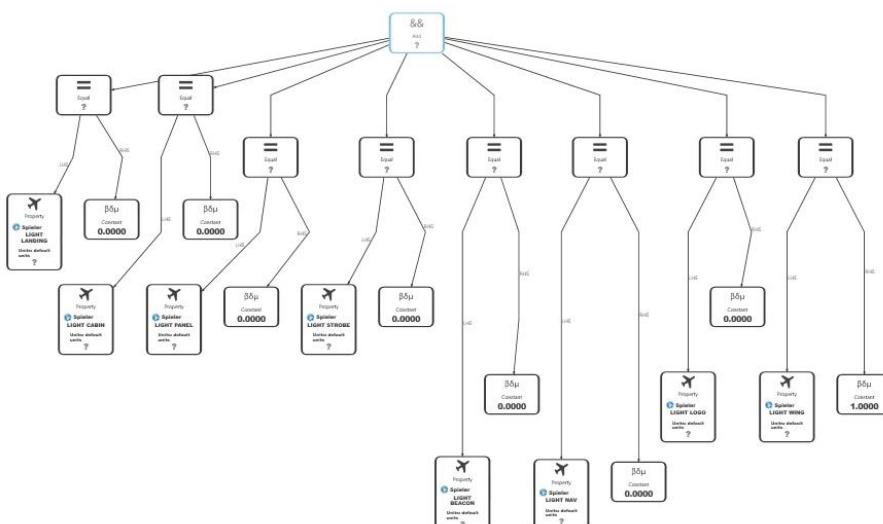


2. Zug: Jetzt wird es kompliziert. Die KI weiss nicht, ob der Spieler, oder die KI begonnen hat. Eine Abfrage, ob Feld 5 der KI gehört ist nicht aussagekräftig, denn der Spieler kann mit einem beliebigen anderen Feld begonnen haben und die KI hat beim 1. Zug nur mit Feld 5 geantwortet.

- Wenn Feld 5 vom Spieler belegt ist, werden die weiteren Züge einfach: Die KI fragt schlicht ab, ob der Spieler 2 Felder in einer Reihe belegt hat und das 3. Feld frei ist. Also wird dieses 3. Feld belegt.

- Wenn Feld 5 von der KI belegt ist, kann es sein, dass der Spieler einen, oder bereits zwei Züge gemacht hat. Darum wird zuerst abgefragt, ob nur 1 Feld (auf den Feldern 1,2,3,4,6,7,8,9) belegt ist.

Dies ist eine von 8 Logiken, hier wird abgefragt, ob Wing (Feld 9) belegt ist und alle anderen 7 frei sind:

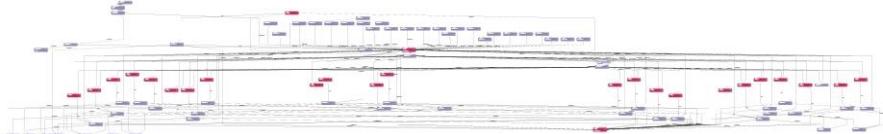


Wenn eine der Logiken anschlägt, wird per Random eines von 2 Antwortfeldern ausgewählt, in obigem Beispiel Feld 3, oder 7.

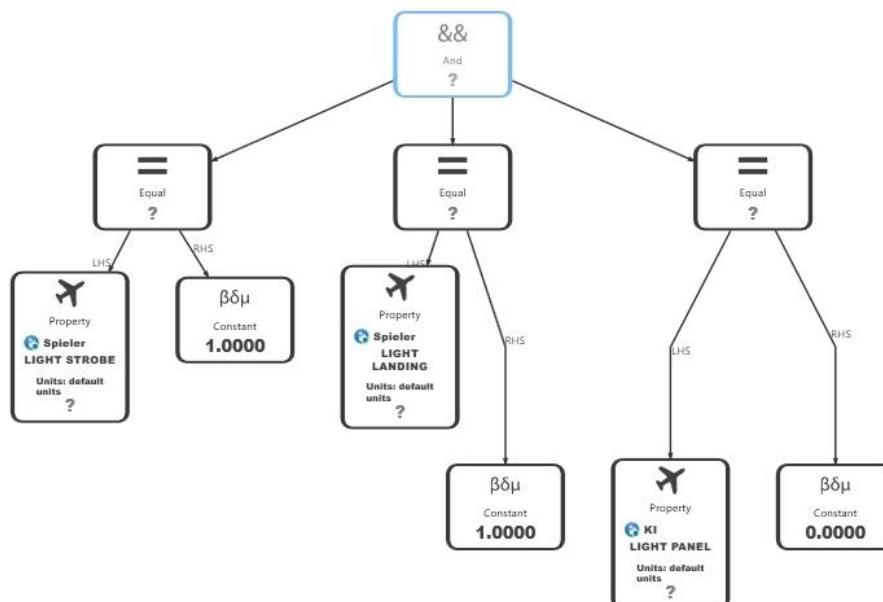
Wenn keine Logik anschlägt, muss man alle möglichen Kombinationen von 2 Feldern (auf den Feldern 1,2,3,4,6,7,8,9) abfragen.

Darum läuft erst eine Abfrage nach 2 vom Spieler belegten Feldern in einer Reihe (Waagerecht, Senkrecht und Diagonal).

Diese Abfrage wird ab jetzt bei jedem weiteren Zug durchlaufen, es sind übrigens 56 Objekte:



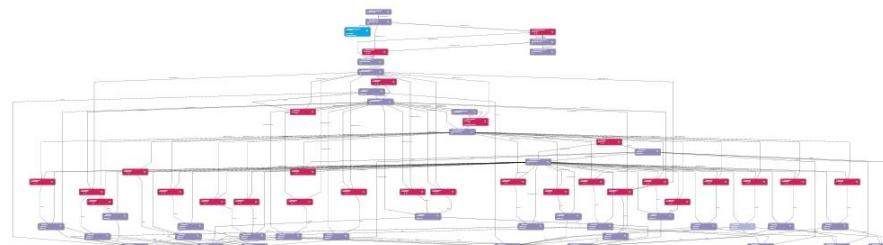
Die Logik sieht so aus. Hier wird eine von 3 Varianten in der oberen horizontalen Reihe abgefragt:



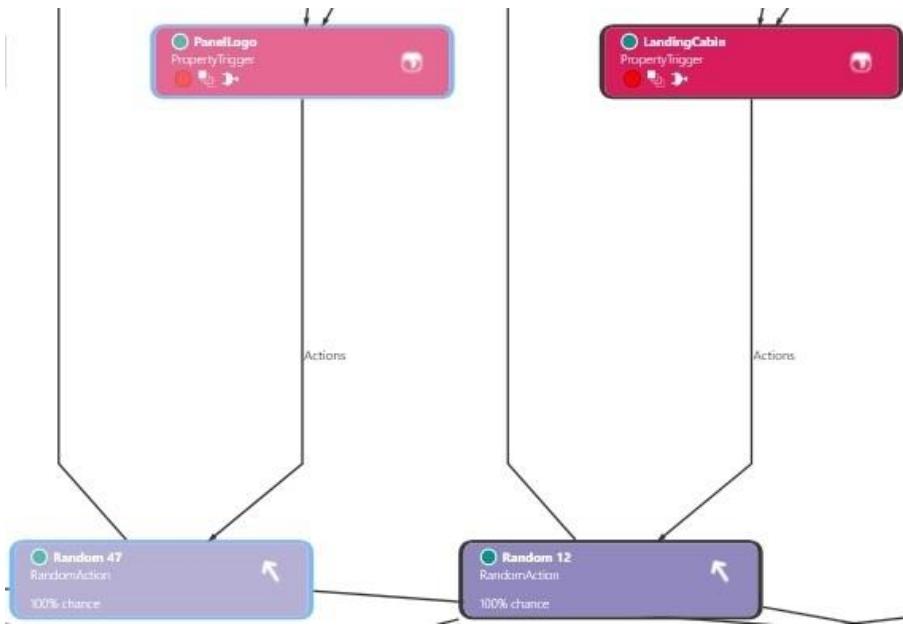
Wenn eine der Logiken anschlägt, wird das 3. Feld belegt.

Wenn keine Logik anschlägt, müssen die übrigen möglichen Kombinationen abgefragt werden, z.b. Feld 1 und 8 belegt, in diesem Fall wird per Random Feld 4, oder 7 belegt.

Diese Abfrage benötigt weitere 56 Objekte:



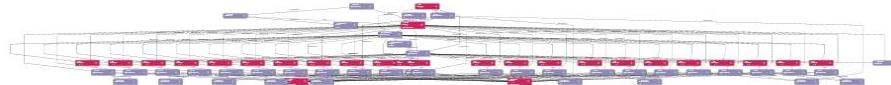
Hier eine Detailansicht:



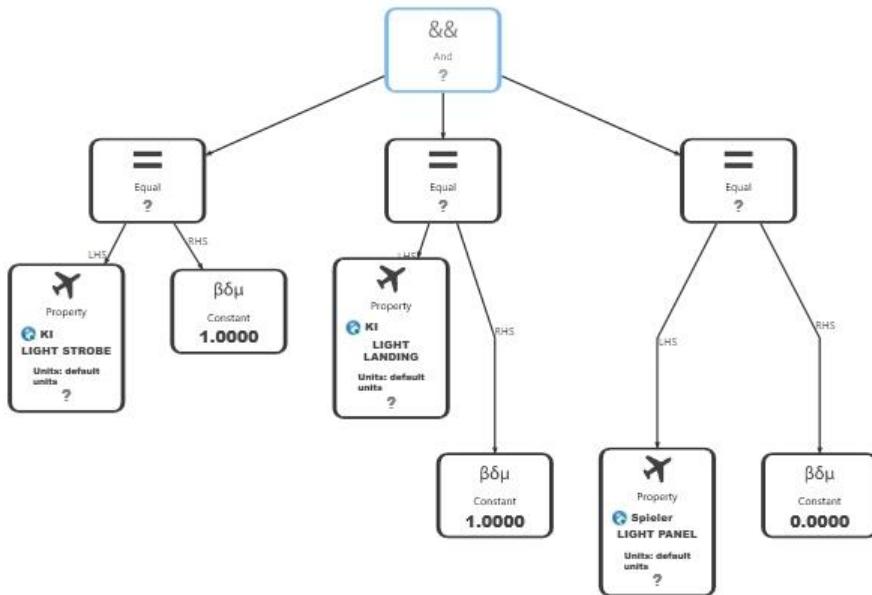
3. Zug: Ab hier wird es einfacher, denn die Züge 3 bis 5 laufen alle gleich ab:

Zuerst wird abgefragt, ob die KI 2 Felder in einer Reihe belegt und das 3. Feld frei ist.

54 Objekte wurden dafür benötigt:



Die Logik sieht so aus:



Schlägt eine der Logiken an, wird das 3. Feld belegt und die KI gewinnt.

Wenn keine der Logiken anschlägt, wird die in Zug 2 schon verwendete Abfrage gestartet.

Sind da 2 Felder vom Spieler belegt, nimmt die KI das 3. Feld um einen Sieg des Spielers zu verhindern.

Wenn keine der Logiken anschlägt, wird die RandomAction der „schlichten KI“ gestartet, also ein beliebiges freies Feld wird belegt, denn ein Sieg ist nicht mehr möglich.

Blieb nur noch 1 Problem zu lösen: Wie Zähle ich die Spielzüge?

Die KI muss wissen, ob sie Zug 1, 2, oder 3 macht, damit die richtige Logik-Folge abgefragt wird.

Man könnte 3 CounterTrigger nehmen, einer zählt bis 1, einer bis 2, und der dritte bis 3. Vor jedem Zug gibt man allen 3 Zählern einen „Tick“ (die Zähler erhöhen sich jeweils um 1), und zünden, wenn der passende Wert erreicht ist.

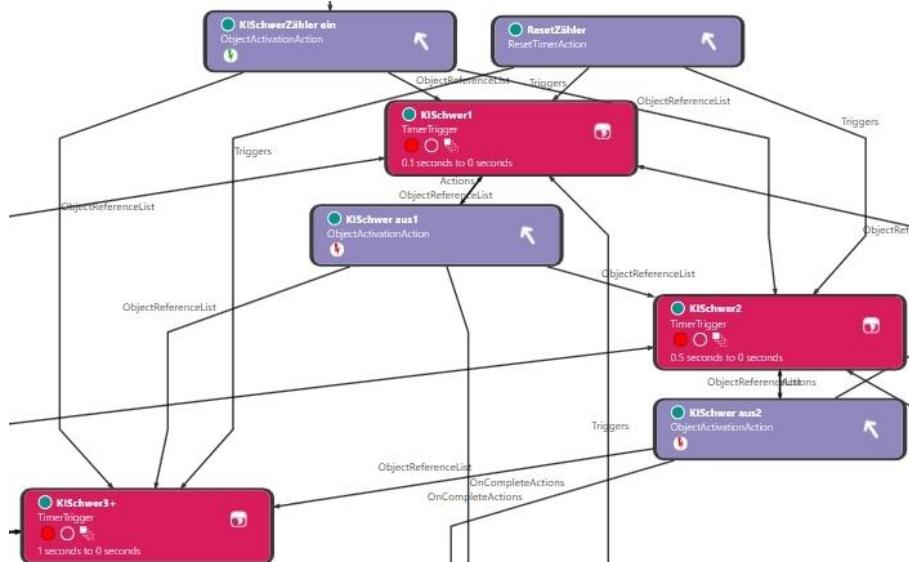
Das Problem bei den CounterTrigger ist, dass man sie nicht zurück auf null stellen kann. Darum sind sie nicht zu gebrauchen, weil nach jeder Spielrunde müssten sie zurückgestellt werden.

Nach langem tüfteln bin ich auf TimeTrigger als alternative gestossen.

3 Timer (einer mit 0.1s, einer mit 0.5s und einer mit 1s) werden vor dem KI-Zug gestartet.

- Der 0.1s Timer startet als Erstes und stoppt sofort alle 3 Timer. Anschliessend werden dem 0.1s Trigger 1000s hinzugefügt, sodass dieser beim nächsten Mal nicht zündet. Zug 1 wird gestartet.

- vor dem nächsten Zug werden die 3 Timer wieder gestartet, dieses Mal zündet der 0.5s Timer, usw.

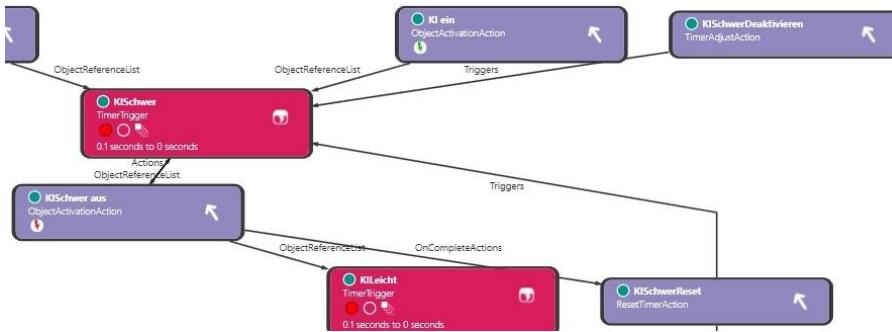


8) Wie unterscheide ich nach dem Spielstart zwischen schlichter und intelligenter KI?

Mit der gleichen Methode wie beim Zählen der Spielzüge (oben), habe ich 2 TimeTrigger Trigger erstellt.

Beide Timer sind auf 0.1s gestellt. Beim Spielstart wird zwischen leicht und schwer gewählt, was zur Folge hat, dass dem jeweils anderen Timer 1000s hinzugefügt wird.

Jedes Mal, wenn die KI am Zug ist, werden die beiden Timer gestartet. Logischerweise zündet der Timer mit 0.1s, dieser schaltet beide Timer ab und resettet sich selber, sodass er wieder bei 0.1s für die nächste Abfrage bereitsteht. Anschliessend zündet er die passende KI (schlicht, oder intelligent).



Mir ist klar, dass obige Beschreibungen nicht komplett sind.

Es fehlen noch Auswertung, wer gewonnen hat, wie die Spielstandsanzeige funktioniert, Einbindung des Goals, usw. Mit diesen Beschreibungen versteht man aber das Prinzip dahinter und wer es genauer wissen will, kann jetzt diese Mission im SimDirector öffnen und analysieren.

Ich hoffe, dir hat dieser Flug Spass gemacht, wenn ja, gib doch bitte eine Rückmeldung an p3d@andi20.ch . Auch Fehlermeldungen (Schreibfehler, falsche Angaben, usw.) an p3d@andi20.ch senden, ich freue mich über jede Rückmeldung.