uSim AI module 1.0

ABOUT: This module helps setting up basic AI behaviour for Aircrafts. The module use a flexible routine based system wich hold the values for the desired flight behaviour of the AI. A serie of routines can be put together and linked to make up an AI action. AI actions are a set of routines that are enabled or processed while the action criteria is matched. Routines hold information for processing vertical (pitch) motion, horizontal (roll/rudder) motion and longitudinal (speed) motion. For example an AI action could be “Fly to waypoint” wich is made by a routine that tells the aircraft to hold waypoint altitude, heading to waypoint and desired speed. The routine can hold a set of triggers to tell what to do after trigger criteria is match.

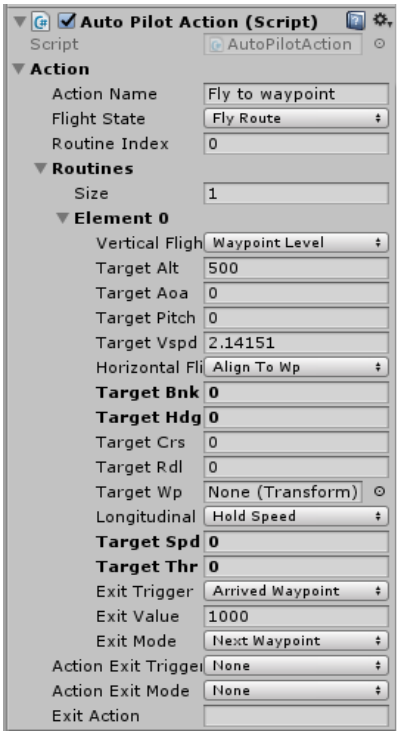
A set or group of actions make a “flight plan”. A flight plan can be setup and saved to be used by any AI flight. Flight plans holds information about origin, destination and aircraft.

OVERVIEW

The module is made of three main systems.

- Flight plan (actions and waypoints for autopilot)
- Autopilot and actions manager (in aircraft)
- Airport systems (aprouches, Ils and taxi)

First let's take a look at how autopilot actons and routines are setup and used. Serval examples for autopilot actions are included.



Here we have an autopilot action wich is named “Fly to waypoint”. The corresponding flight state of it is Fly Route. The action holds just one routine wich is set to fly the waypoints.
Action name is used to identify the action in the flight plan while flight state is used to identify the type of action. Both are used to fetch and set an action for autopilot.

Actions also can have exit criterias or triggers. An action exit trigger can be selected and the corresponding exit action can also be selected. One common situation would be to have action exit trigger set to “arrived final waypoint” and exit mode set to “Exit action”. So in exit action value we can put the name of the next action to load in the autopilot.

Let's take a look at the routines.

Routines hold directives for the 3 main axis of flight. Vertical flight, turning and forward motion.

Vertical flight is tied to the elevator control while horizontal is to the aileron and rudder control. Longitudinal is tied to engine throttle.

Each axis has a serie of target values that are used depending on the mode they are set to. Some modes overwrite some target values. For example in this case, “Hold speed” target is set from the autopilot aproauch speed configuration.

In rare ocations this values need to be setup on the routine.

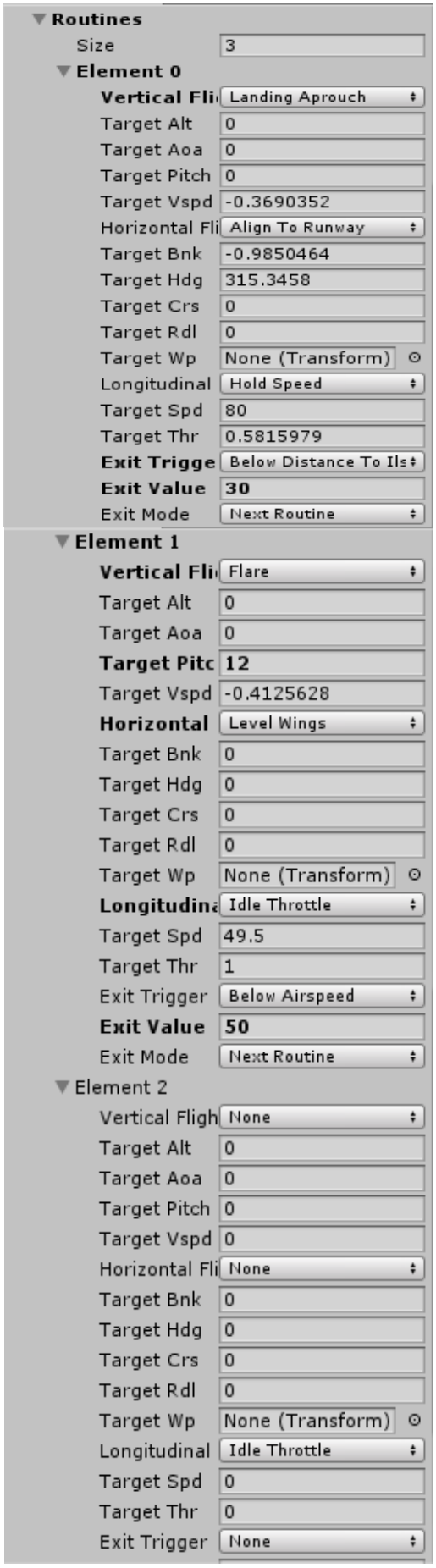
Routines can also have exit triggers to either enable the next routine in serie or exit.

In the example shown when distance to runway ils is below target distance of 30 mts, next routine is triggered wich consist of the “flareing” instructions.

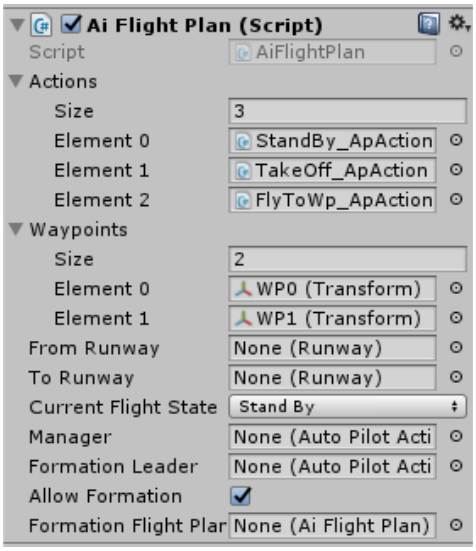
Also as speed drops below 50 a next routine is triggered wich sets throttle to idle.

So the action makes the aircraft aprouch, flare and then stop.

Next we'll take a look at Flight plans and how they are setup.



Actions can be saved as prefabs and therefore reused in flight plans just by droping them into the actions array.

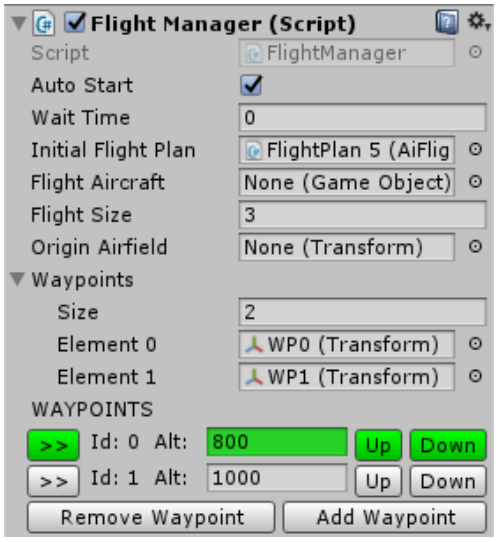


This example flight plan is typically used to make an aircraft takeoff and fly waypoints.
Also waypoints can hold custom triggers that can link other flight plan or to trigger an aproauch.

Waypoints are feeded from the flight manager editor wich we will see later.

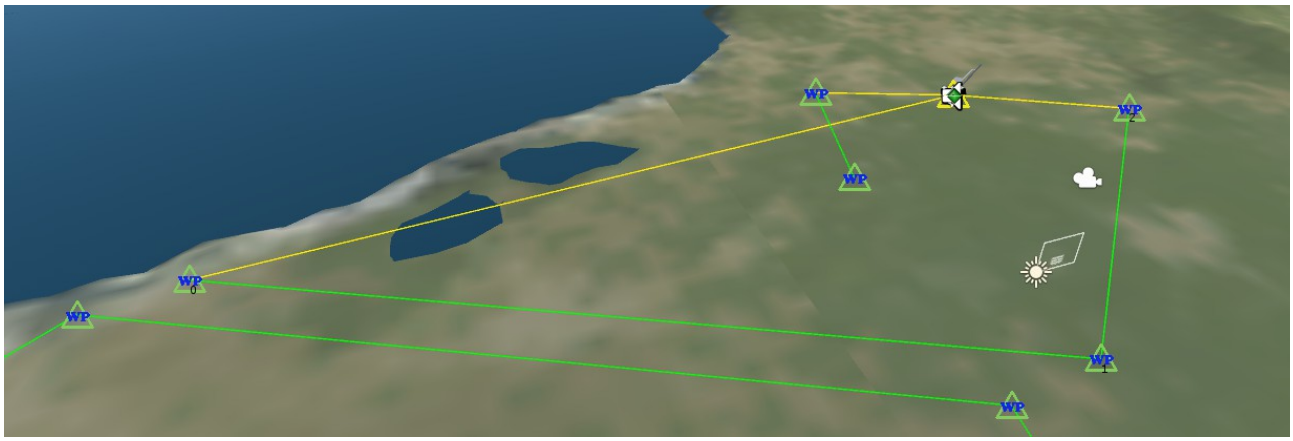
Most other values are handled by the flight manager as well. All left to adjust here is if the flight can allow formation flying when the flight is of more than one aircraft. And if it's tilded then you need to assign a formation flight “flight plan” this is included in the flight plan prefabs folder.

Flight manager

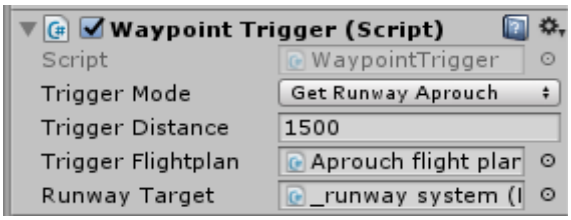


This is an editor tool and also the object to hold all the flight information.

AutoStart: if set to true, the system will wait “wait time” to start the flight. On start, if the aircraft is on the runway it will set the flight mode to takeoff. If it's far from runway it will fetch the taxi info from the origin airfield. If no taxi info is found the flight will just stay on stand by.
InitialFlightplan: this is the main flight plan of the flight. This is ussually the action that fly around the waypoints.
FlightAircraft: The aircraft this flight is assigned to.
FlightSize: the amount of aircraft to be spawned as wingman.
OriginAirfield: The runway system object from wich it will depart.
Waypoints: remove waypoint, deletes the one currently selected. Add waypoint adds a new waypoint after the selected.
Up and down allows changing the order.
When the flight manager is selected the waypoints and flight path is shown in the inspector



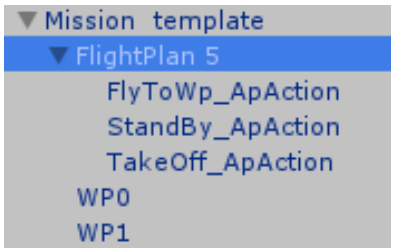
if the final waypoint has a waypoint trigger set to get a runway aprouch, then a yellow line will link the waypoint to the desired runway.



This is how a waypoint trigger attached to the final waypoint looks like.
When the mode is set to “Get runway aprouch” the aprouch flight plan is get from the target runway.
When the mode is set to “use Trigger flightplan” the autopilot system will load the trigger flightplan set in “Trigger Flightplan”.

This items make up the flight plan. Flight plans are stored on scene and the asigned autopilot will clone the flight plan and keep it in the aircraft hierarchy for acces.

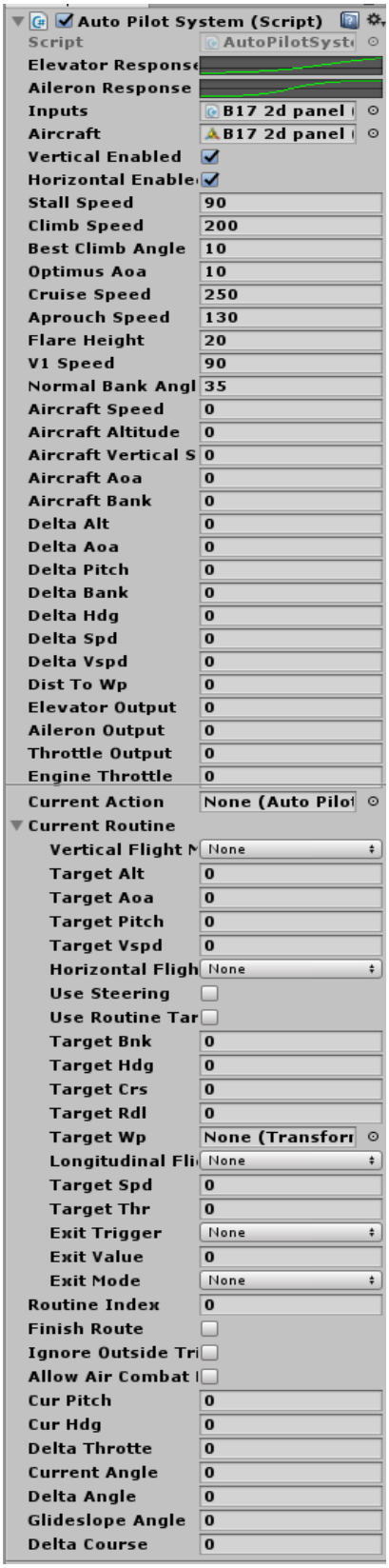
This is how a flight plan prefab looks like.



Now let's take a look at the aircraft side of the AI system.

Autopilot system

The aircraft needs to have two scripts wich work in tandem to get the AI flying properly. One is the autopilot itself wich is resonsable for generating controls output and an actions manager that handles the state of the flight and feed the autopilot.
Let's see what each field does.



Elevator response: the amount of elevator input in function of target pitch in degrees.
Aileron response: the amount of aileron input in function of target bank in degrees. Hold heading function internally would adjust banking based on desired heading.
Inputs: The uSim inputs manager of the aircraft. Drag and drop the aircraft object into the field from scene.
Aircraft: the aircraft rigidbody. Same as before.
Vertical and horizontal enabled are used to toggle the vertical and horizontal axis on/off.

Aircraft reference values

Stall speed: used for flareing instructions and take off.
Climb speed: the target speed for climb out
Best climb angle: the angle to hold during climb out
Optimus Angle of attack: it's not currently used.
Cruise speed: the speed to cruise trough waypoints.
Aprouch speed: the speed the aircraft use for aprouches to runway
Flare height: the height the aircraft will attemp to hold above runway until speed drops enough to make touch down.
V1 speed: the speed at wich the aircraft will rotate into the climb out.
Normal back angle: the max banking angle used in turns.

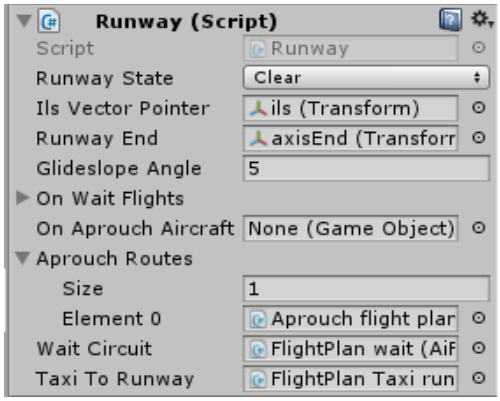
The rest of the fields are read outs and doesn't need to be filled. This values are handled internally.

Note that at this moment air combat manouvers are not implemented.

Then an action manager script needs to be also attached to the aircraft but this script doesn't need any adjusting as it is the one to handle the comunication between flight plan and autopilot.
If this script is turned off then the autopilot can be manipulated manually.
To do so you can change the autopilot axis modes and set the proper target values.
For example setting horizontal axis to hold bank and setting the bank would make the aircraft turn. Setting it to hold heding can be used to hold a desired heading. This can be usefull to add autopilot functions to the player aircraft. Althou it's not yet fully supported and there's no runtime ineraction.

Now let's see how airports are setup to work with the AI.

Setting up Airports



Airports are made of runways and taxi paths.
First we will take a look at the runway object.
This class holds all the information relevant for the autopilots to work with runway aprouches and landing as well as traffic awareness.
Runway states are three.
Clear: It can be used by aircrafts to either takeoff or land. Landing and taking off set's the runway to...
Traffic: It means that there's an aircraft eitehr taking off or on final aprouch.
Finally, inoperative is a disabled state althou it's not implemented in any case yet.

Ils vector pointer: the ils transform that is used as reference for the direction of the runway and runway glideslope end.

Runway end: a transform used to set the runway axis from ils to runway end.

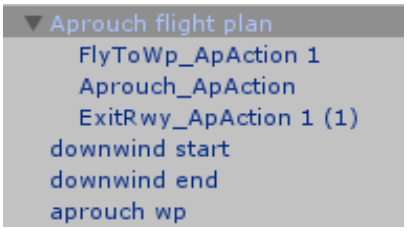
Glideslope angle: the angle in degrees of the aprouch glideslope.

On wait flights: holds the aircrafts that are currently flying the wait pattern.

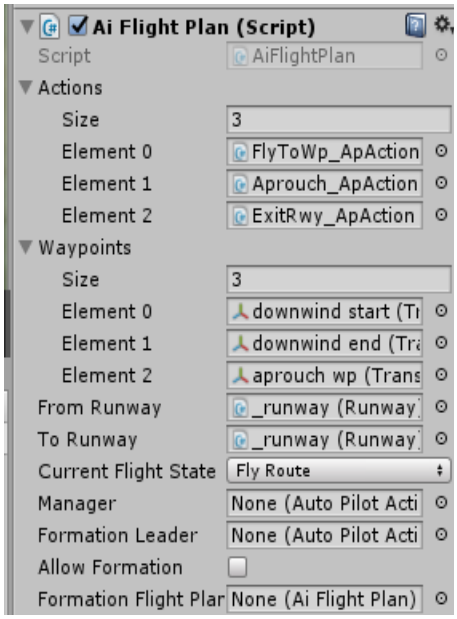
On aprouch aircraft: the aircraft currently on aprouch and that has triggered the runway traffic awareness to “traffic”.

Aprouch rutes: Airports use pre stablished flight-plans like normal flights. This flight plan holds the actions and routines for the autopilot to follow upon triggering an aprouch.

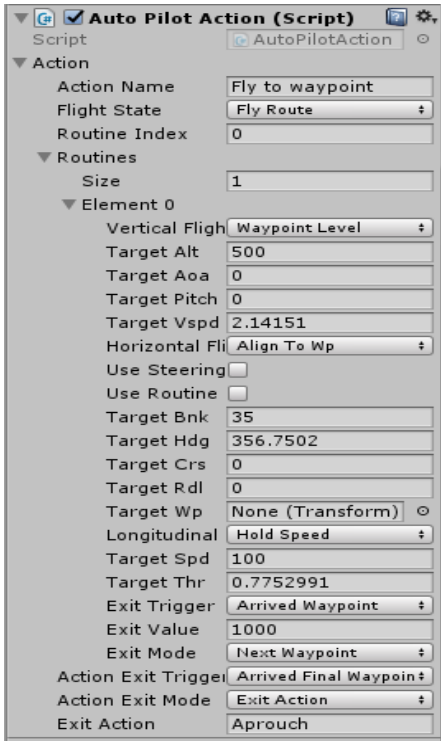
This is how a typical aprouch flight plan looks like.



As seen in the pictures the flight plan has a serie of actions that leds the plane into the runway via a set of waypoints.
In this case a pattern with downwind start and end leads into a base leg to intercept the aprouch waypoint.



This is how the fly waypoints action of the aprouch rute looks like. The routine travels trough the waypoints and when arriving final waypoint the exit action “Aprouch” is loaded. (wich is the second action of the aprouch flight plan).
NOTE: the flight plan needs to have current flight state to “fly route” for the aircraft to continue the flight.

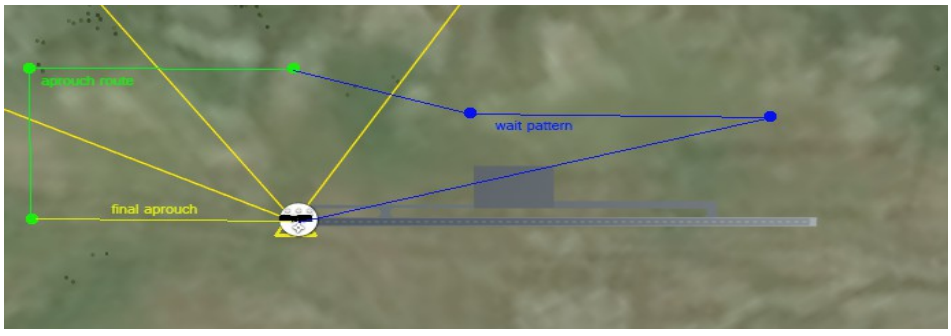


In previous items we described how waypoints trigger are used to link flight plans to aprouch routes.

When entering the final aprouch action, if the target runway is on traffic, then the autopilot will load the “wait circuit” flight plan set in the runway class.

Wait patterns work same as normal waypoint flying. The final waypoint need to have a waypoint trigger to link back to the aprouch rute like normal flight plans do.

If runway is again on traffic when the aprouch waypoint is reached then wait pattern get's loaded again, and so on.



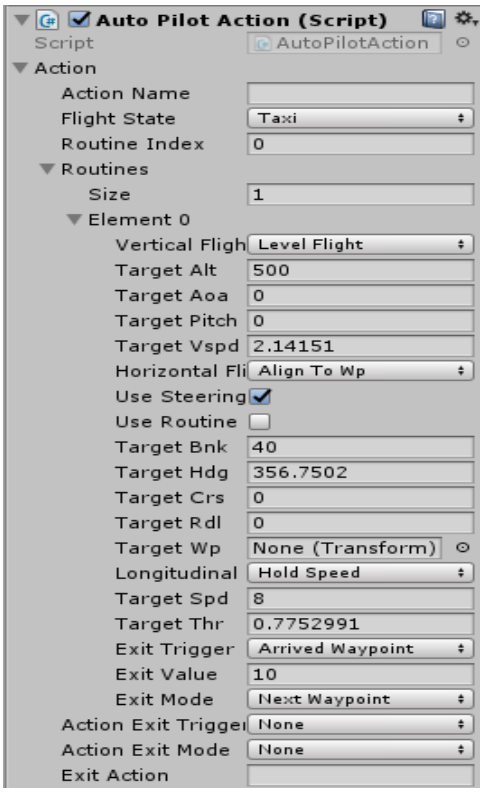
Taxing

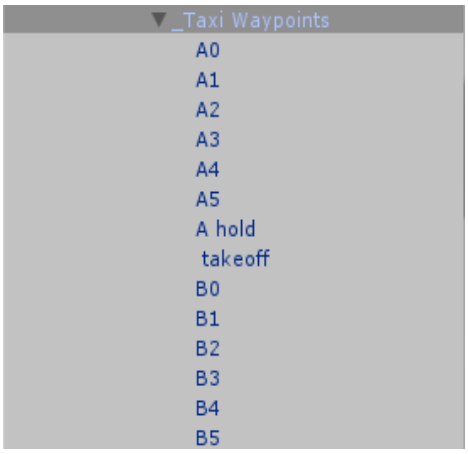
Taxing is nothing more than a “fly waypoints” routine but with out vertical axis action and with target speed set to taxing speed. Ground handling of the aircraft is enabled by checking “use steering” option in the action.

Flight state is need to be set to “Taxi” for the autopilot to internally handle runway traffic actions. The taxi flight plan needs to have a “stand by” action that is used when the aircraft needs to hold position during taxi.

Autopilot do this by setting the flight state from “Taxi” to “Stand by”.

NOTE: is important to have in mind that when exit trigger is set to arrived waypoint, exit value is the distance in wich it is triggered.





Taxi waypoints are used by taxing flight plans.

Each runway generally need at least two taxing routes. One for parking to runway, and other from runway to parking.

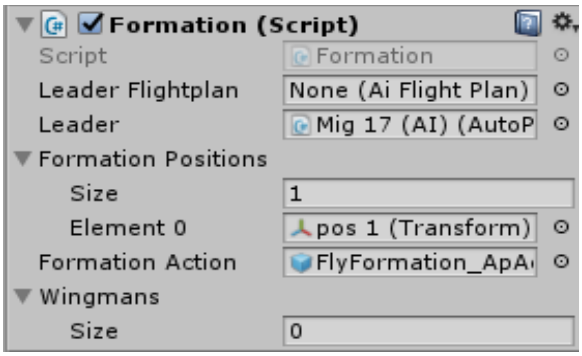
Finally, waypoints triggers are used to handle holding the runway, triggering takeoff and triggering parking.

See taxi waypoints in the example to understand further how triggers are setup.

Wraping up

Those are basically the three systems that work in conjuntion to have civilian aviation type flying. Optionally flight plans can handle formation flights. Next we will take a look at formation system.

Formation class is a simple configuration script to be used by formation flight.



Basically add a formation object prefab into the aircraft hierarchy and set three items:

leader: the aircraft it is attached to.

Formation poistions: this should be allready set. It's the transforms used by wingman to hold the formation poistion.

Formation Action: the action that is used to hold the formation. (prefab included)

for each wingman spawned a position is asigned, so you need same or more number of positions than the flight size.

To enable the formatoin spawning the flight manager needs to have the flight size set to more than 0 and the flight plan needs to have “Allow formation” enabled.

NOTE: if the formation is starting from the ground the wingman will be spawned at the formation position on ground. So runways may not be suitable for flights of more than two aircrafts. At least for now.

The example scene has a flight of two mig 17 taking off from runway.

Let's setup a flight of 3 B17s for the demo scene.

Setting up AI flight

Aircrafts can start on the ground but also can start flying. In this case we will setup a flight of 3 b17s that are allready flying into the runway for joining the aprouch rute.

First we need to setup the aircraft with the AI autopilot and actions manager. Then we need to fill the autopilot reference data for the aircraft.

Also since the aircraft is AI controlled, we need to turn off the vehicle camera. Look for “vehcam” object and turn it off.

The scene allready has a p51 starting on the air and performing the same intended action so we can just clone this flight plan.

By doing so waypoints and flightplans are cloned and porperly duplicated. So all we need to do is assign our b17 to be the “flight aircraft”. Do so by draging and droping the aircraft from scene into the field.

The diference in this flight is that the p51 flight doesn't have formation. So we need to add the formation object to it.

drag the formation prefab into the aircraft as child of it. Assign the aircraft to the leader field. Add a new position transform for the second wingman and add the position to the formation positions array.

Since it's starting in the air we need to set the aircraft control script aircraft speed to something around 180 or so. This speed is set in knots. Also we set gear down to false.

Also we need to have the engines running. To do so, set in the aircraft InputsManager the field engine on to true;

Now in the flight manager we set the flight size to 2.

NOTE: Formation flying and combat manovering is still under development and will be polished and added in future releases. Some aircraft may experience unstable formation flying with out proper aircraft response configurations.

Formation flying state has a position and direction lerping routine with helps the wingmen to hold the position. If the aircraft is too far at the moment of switching into formation flight, it might experience some warping until the distance is closer.

That should be all. For more information and assist on setting up AI behaviours in your scene feel free to contact via mail at gabriel.campitelli@gmail.com

Thank you!
Gabriel Campitelli

uSim developer.