Creating the MAAS

Upon a user request on armaholic I started to look into creating an Arma 3 mod for a mobile aircraft arresting system (MAAS). This report shall describe the different steps on the way to achieve that goal. It is currently a work in progress, as the work is far from complete.
On expeditionary airfields mobile aircraft arresting systems can be deployed to help planes landing. The system is composed of two trailers and a wire deployed between them. A planes tailhook can grab the wire and brake more rapidly.

First steps

As maintainer of the Nimitz I already have access to scripts and a model where wires are used for stopping planes. I use these as a reference for the new system. As there is no suitable trailer available in Arma 3, a new one needs to be created. I reach out to several modellers and see if anyone is interested in the project. Before I do that I assemble a first prototype to get out of the idea stage quickly.

Prototype

For the prototype I decide to use existing arma assets and a custom wire model. The wire model is very simplistic, merely a stretched cube with a length of 40 meters, textured with the black procedural color. It contains mem points so it can be animated, e.g. the tailhook shall pick up the wire and stretch it. Instead of the trailer I use a pair of Huron containers that can be airlifted for now. It's not clear in my mind how the deployment of the MAAS will be in game, either place it in Editor only, or place it via action menu entries or maybe via a defined key press sequence. With that in mind I create a model.cfg for the wire so it is animated. The animations are of type *translationSomething* so they alter the wire in the three dimensions. A sample is provided here:

```
class ani_cable_Wide
{
	type = "translationX";
	source = "user";
	memory = "true";
	minValue = -35;
	maxValue = 20;
	axis = "cable_wide";
	selection = "wire";
};
```

This entry is a copy of an entry in the existing model.cfg for the Nimitz wire system. Now the MAAS addon needs the config.cpp file, so it can be found in game. The config.cpp is composed of several sections for this project: CfgPatches, CfgEditorCategories, CfgEditorSubCategories, CfgVehicles and later CfgFunctions.
The CfgPatches sets up the addon:

```
#define _ARMA_

class CfgPatches {
	class ttt_maas {
		units[] = { "ttt_maas_wires" };
		weapons[] = {};
		requiredVersion = 1.62;
		requiredAddons[] = {"A3_Structures_F"};
		author = "TeTeT";
	};
};
```

The *requiredAddons* refers to a basically arbitrary Arma 3 pbo which is dependent upon. Tools like Mikeros pboproject can build the requiredAddons array dynamically, based on dependencies found in the addon.

```
class CfgEditorCategories {
	class TTT_MAAS_obj {
		displayName = "MAAS";
	};
};

class CfgEditorSubcategories {
	class TTT_ec_wires {
		displayName = "Wires";
	};
};
```

These two sections define where in Eden the MAAS wires will be found. Eventually there will be trailers as well.

```
class CfgVehicles {

	class ThingX;   // External class reference
	class ttt_maas_wires: ThingX {
		scope = 2;
		scopeCurator = 2;
```

```
scopeArsenal = 1;
vehicleClass = "Tents";
displayName = "Wires";
model = "\ttt_maas\wires.p3d";
editorCategory = "TTT_MAAS_Obj";
editorSubcategory = "TTT_ec_wires";
destrType = "DestructNo";
cost = 100000;
```

The wire itself is setup as a ThingX and belongs to the Tents vehicleClass, which is a bit arbitrary but works.

```
class AnimationSources {
	class ani_cable1_long {
		source = "user";
		animPeriod = 1e-006;
		initPhase = 0;
	};
```

Here the animations are exposed to the user space, so I can use SQF to animate *ani_cable1_long.* There are many more entries like this, as I first went with a four wire setup like found in the Nimitz. Later this will be reduced to a single wire.

```
class EventHandlers {
	class CBA_Extended_EventHandlers: CBA_Extended_EventHandlers {};

	init = "this execVM 'ttt_maas\functions\fn_init.sqf';";
};
```

Here the scripts that control the wire will later tie in. I've used some boilerplate code for CBA support, as I probably will use CBA key press events for setting up deployment of the MAAS. Eventually I will add CBA as requirement to requiredAddons in CfgPatches. For a start I call the initialization script via execVM. Further down the road the init script will be placed in CfgFunctions and executed via spawn.
Probably a surprise is, that the fn_init.sqf right now contains no code. At this stage of the project I feel that it is rather cumbersome to place code directly in the addon - I rather develop sample missions that illustrate the system. A mission is much more flexible when coding than an addon, as addons need to be build outside of Arma 3 and the game needs a restart before testing a new version is possible.

Mission Wire Animation

The sample missions shall illustrate the working of the system and test out in a bottom up approach what is possible with the engine. So first I test if the animations work. For this I place a F/A-18E named *plane* on the runway and a wire named *maas* on the runway. The following code attaches the animated wire to the tailhook from Init.sqf:

```
[maas, plane] spawn {
	params ["_maas", "_plane"];
	private ["_thPos", "_pointPlane", "_wire",
		 "_dWide", "_dLong", "_dHigh"];

	_wire = _maas selectionPosition "wire2";

	_plane animate ["tailhook", 1, true];
	_plane setVelocity [0,0,-0.5];
	sleep 1;
	_thPos = _plane modelToWorld (_plane selectionPosition "tailhook");
	_tailhookAnim = _plane animationPhase "tailhook";
	_pointPlane = _maas worldToModel _thPos;

	_dWide = (_pointPlane select 0) - (_wire select 0);
	_dLong = (_pointPlane select 1) - (_wire select 1);
	_dHigh = (_pointPlane select 2) - (_wire select 2);

	_maas animate ["ani_cable2_wide", _dWide];
	_maas animate ["ani_cable2_long", _dLong];
	_maas animate ["ani_cable2_high", _dHigh];
};
```

This simple mission validated that the wire animations work on my wire model and can be controlled via the *animate* SQF command. Now I take the existing arresting code from the Nimitz and modify it so it uses the new wire model. For this I use two functions, *ttt_maas_fnc_useWires* and *ttt_maas_fnc_wires.* The first one brakes the plane down and calls the second one, which is responsible for animating the wire.
I'll test the mission and share it with the Nimitz development team for early feedback.

Mission Deployment

In the first experimental mission it was shown that the animated wire system and braking system does work. The question is now, if and how players can operate the MAAS in game. Right now

it can be deployed only from the editor, but it would be nice to place it dynamically. So I write another small mission that explores this option. I need two functions for this, ttt_maas_fnc_deploy and ttt_maas_fnc_pack. The first deploys the MAAS and the second packs it up. As I have no custom trailer objects yet, the deployment script will be tied to a HEMMT box vehicle as action menu entry. Likewise Huron containers will be used as placebo object for the trailers.

Outlook

The scripts from the experimental missions need to be embedded into the addon. The code quality needs to be improved too - the missions are written quite sloppily to get a result quickly.
Once the custom trailer object is there, it needs to be integrated into the scripts as well.
When the final ingredients are there, the system will be tested and examined and eventually published.

Resources

You can find the sample missions and pbo at
http://tetet.de/arma/arma3/nimitz/experimental/maas/
Please do not mirror, the system is by far not ready yet.