The Special Aircraft Service presents…

SAS AI, Engines, Hotkeys and Carriers Mod v2.5 for v4.101m

User Guide

About

This is not an AI mod per say, instead this is a collection of over a dozen related mods which alter engine behavior, add additional hotkeys, new aircraft features and most of all, tweak the default AI routines. For a while, many of these mods weren't compatible with each other as they all used the same core class files. This compilation has taken away that difficulty and combined them all in one, easy to use pack.

Now standard in the DBW compilation (from v1.7 onwards), it is hoped this pack will one day be part of the standard ModAct and help achieve the dream of a unified modding code base. For the end-user, it means expansion of the games features and access to new elements such as air-to-air refueling, fuel dumping, realistic air start procedures, carrier operations and eventually, radar and modern electronic warfare.

It is a continual work in progress and is be updated to allow for new features. Currently it is only compatible for 4.101m, but in future will be made compatible with 4.11m (albeit minus Cert's AI component).

Installation Instructions

For DBW Users

1) This pack is virtually identical to what is included in DBW1.71 already. There is a minor tweak to gunner behavior. Otherwise the rest is the same
2) To install, add to your #DBW directory
3) Enjoy

For ModAct 4.101m Users

1) Extract to your #SAS or MODS directory
2) Enjoy

Features

AI

- Certificate's AI mod, coupled with a few additional tweaks, offers a brand new gameplay experience in single-player missions by breaking up the previous monotonous AI routines. Based on skill set, AI pilots will demonstrate a different degree of maneuverability and difficulty. No more AI planes outrunning you and climbing endlessly to the moon. In addition, some tweaks have been done to improve ground detection and stop crashing during low-level maneuvers and landing. This mod will be superseded by Team Daidalos's code in the 4.11m release.
- AI Overheating and Supercharger mod is self-explanatory. AI will now experience engine overheating and they can now properly switch supercharger stages.
- The night-fighter tweaks allow aircraft with Schrage Musik to attack aircraft from below. The Bf-110G-4 and He-219UHU use this as their default attack pattern, but can switch to a zoom and boom pattern if better suited. This feature will be altered in 4.11m to use the new code.
- Ground Attack Mod + Orders improves the way AI attacks ground targets, in addition to allowing them to sight them earlier
- Sniper Gunner fix prevents AI from targeting the player with perfect accuracy. In addition the turret rotation speed is reduced and the detection range is significantly reduced at night time

Engines

- As previously described, AI now suffer from overheating and now properly use Superchargers
- More realistic engine start-up/shutdown/windmilling behavior. This code is now designed so specific engine types will be a bit more characteristic of their real-life counter-parts. Radials feature more misfires and may need restarting whilst big inline engines will have long shutdown/windmilling times compared to their smaller counterparts. Also the normal windmilling behavior has been altered, preventing the previous unrealistic complete shutdown on gravity-feed carbureted aircraft in temporary G's (though excess negative G's will cause an eventual starvation)

- Added two new engine types: Rotary and Turboprop. The Rotary engines feature hand-starting, long wind-down times. With future tweaks, we hope to model the ‘blipping’ and mixture controls more realistically. The Turboprop is still WIP but may be used as a placeholder (for now, it is a modified clone of an Inline engine).
- Full Throttle Mod is included, though the smoke effect has been reduced significantly to more realistic (and better performance) levels.

Hotkeys

- Based on the old ‘Bomb-bay Doors’ mod, this new version of Aircraft Hotkeys is significantly expanded. In addition to the bomb bay doors mod, there are now multiple new hotkeys for features including fuel dumping, weapon salvo size, delay time etc. A full listing is included below
- Differential braking has been added as well as passive wheel steering (more instructions included below for modders)
- Flap mod added, allowing modders to assign addition flap positions to aircraft beside the 3 defaults
- One of the main new features is the 'Weapon Salvo Size' key, which allows you to define the salvo size for rockets and bombs. On aircraft which fire in pairs, it offers 3 modes: single fire, pair fire and full salvo. For aircraft that fire in salvos by default, the modes are: half normal salvo, normal salvo, all salvos (e.g. if normally 12, it’s now 6, 12 or all). In addition you can define the delay between ordnance using the 'Weapon Release Delay' key. You can set a delay between 125msec to 1sec, which functions in single fire mode (when held down) and for large bombers, during automated release. These features were inspired by ZloyPetruskO's mod.
- Fuel dumping has been changed. For modders, they can define if an aircraft can dump fuel and using the ‘Dump Fuel’ key, allow release of fuel from predefined valve locations. In game, if the selected tank (defined in code) has fuel, the valve will open and fuel vapor will be seen externally. When empty, fuel will automatically stop flowing or you may close valve when desired amount is reached.
- Drag chutes have been added for new aircraft. To deploy, press “Deploy Drag Chute” key. If you are above a pre-determined speed (usually >600kmh), it may rip off. To release, press “Deploy Drag Chute” again. It may only be deployed once. Currently features on F-86D-45 and F-86K (soon MiG-21PFM).
- Legacy features from original Bomb Bay Doors mod are mentioned at the end (under ‘Advanced Settings’), along with the ‘just for modders’ section
- PAT Smoke included too

Carriers/Catapults

- This section needs no introduction. This mod contains a significantly' updated version of Fireball's Carrier and Catapult, given a new treatment by western0221 and Benitomuso.
- In addition to improving AI carrier behavior and enabling catapults, this mod now sets custom catapult settings for multiple carrier types, increased arrestor wire strength and has code for future steam-powered systems
- Please see 'Advanced Settings' for more details.

Advanced Settings

I.e. details pertaining to some of more advanced features of this pack. Some basic modding knowledge is advised. These setting guides are possibly out of date and may be redundant. They will be updated at a later stage.

Original Read-Me for 'Bomb-bay doors' Hotkeys mod with some updates:

Based on the hard work of Zuti and Fireball (BombBayDoors Plus 2.5.3)
New additions and compilation by SAS~Anto

IMPORTANT NOTE: Depending on which options you choose, this mod may involve adding lines to the conf.ini file in your IL-2 game folder. BEFORE YOU MAKE ANY CHANGES TO THE CONF.INI FILE, MAKE A COPY OF IT, just to be safe.

The mod lets the user or server modify the way several controls work. You can pick and choose which of the options you use. Here's a summary

- Manually open/close bomb bay doors.
- Manually open/close side cockpit door on Spitfire XII and XIV.
- Display TAS in addition to IAS in the speedbar. (also requires the HUDConfig mod)
- Separate gear up/down controls.
- Separate tailhook up/down controls.
- Separate radiator open/close controls.
- Toggle music on/off while flying.
- Modified airshow smoke.
- New placeholder keys for development of advance jet aircraft (e.g. weapon systems, countermeasures, radar)

Instructions:

By default this mod will allow you to manually control bomb bay doors, side hatches and fuel dumping. The rest of the features must be enabled by adding lines to the [Mods] section of the conf.ini file in your IL-2 1946 game folder, or the [Mods] section of the mission (.mis) file if it is a mission parameter. See the Notes at the end of this README for more details.

Manually open/close bomb bay doors

With this option, whatever key you assign to 'Bomb Bay Doors' key in your controls becomes a toggle to open/close the bomb bay doors on aircraft that have bomb bay doors. When this option is set, you can only release your bombs after opening the bay doors. It has no effect on aircraft that do not have bomb bay doors. This option is turned on by default. To disable it, in the [Mods] section of your conf.ini add the line: BombBayDoors=0

Manually open/close side cockpit door on Spitfire XII and XIV

With this option, whatever key you assign to 'Power 40' in your controls becomes a toggle to open/close the side cockpit door on the Spitfire XII and XIV. This option is turned on by default. To disable it, in the [Mods] section of your conf.ini add the line: SideDoor=0

Display TAS in addition to IAS in the speedbar

This function also requires the HUDConfig mod. With this option, when you cycle through your speedbar you will have three additional settings: kmh TAS, knots TAS, and mph TAS. In the [Mods] section of your conf.ini add the line: SpeedbarTAS=1

Separate gear up/down controls

With this option, your Toggle Gear control only acts as Gear Down, and whatever key you assign to 'Magneto Previous' is Gear Up. In the [Mods] section of your conf.ini add the line: SeparateGearUpDown=1

Separate tailhook up/down controls

With this option, your Toggle Tailhook control only acts as Tailhook Down, and whatever key you assign to 'Magneto Next' is Tailhook Up. In the [Mods] section of your conf.ini add the line: SeparateHookUpDown=1

Separate radiator open/close controls

With this option, your Cowl Flaps (radiator) control only opens your radiator, and whatever key you assign to 'Power 30' closes your radiator. In the [Mods] section of your conf.ini add the line: SeparateRadiatorOpenClose=1

Toggle music on/off while flying

With this option, you can turn the Takeoff and In-flight music on and off with whatever key you assign to 'Toggle In-Flight Music'. You can't turn off the Crash music because...well...because your dead. In the game, as in life, you'll find there are lots of thing you're no longer able to do once you're dead. In the [Mods] section of your conf.ini add the line: ToggleMusic=1 (enabled by default)

Modified airshow smoke

With this option, the normal wingtip smoke is moved to a single smoke stream coming from the middle of the aircraft, and you have the choice of red, white or blue smoke. To enable, add the following line to your conf.ini: AirShowSmokeEnhanced=1 (disabled by default)

To change color of the smoke, add one of the following lines:

```
AirShowSmoke=1     <----- red smoke
AirShowSmoke=2     <----- white smoke
AirShowSmoke=3     <----- blue smoke
```

Notes

- After adding a [Mods] section to a mission file, if you later edit and save the mission in the FMB, the [Mods] section will go away. You'll have to manually add it again with Notepad or other text editor.

- You can use the samples below to copy/paste the lines for any option you choose. Depending on which mods you already have installed you may already have a [Mods] section in the conf.ini or mission file. If so, just add the lines to the existing [Mods] section. Notice a slight difference in the format of parameters in the two files. In the conf.ini, the parameter name and value are separated by an '=' sign, whereas in the mission file the name and value are separated by one or more spaces.

Sample [Mods] section for conf.ini:

```
[Mods]
BombBayDoors=0     <----- This will DISABLE the manual bomb bay door control. It is enabled by default.
SpeedbarTAS=1
SeparateGearUpDown=1
SeparateHookUpDown=1
SeparateRadiatorOpenClose=1
ToggleMusic=1
AirShowSmoke=3
DumpFuel=1
```

Sample [Mods] section for a mission file:

```
[Mods]
  ExternalViewLevel 2
  ExternalViewGround 1
  ExternalViewDead 1
  PadlockLevel 2
  ExternalPadlockLevel 1
```

- Although it is not required for the TAS speedbar displays to function properly, you can modify 'KIAS' to read 'knots' on the speedbar by changing the 'SPD.gb' parameter in 'files\i18n\hud_log_ru.properties'.

Original Read-Me from Carrier Take-off+Catapult mod with some updates:

Originally by Fireball. Updated by western0221 and benitomuso.

- The way the catapult works is that you release your chocks and taxi up to the start of the catapult track. When you're close, within about 10 meters or so, stop and put your chocks back in. If you're close enough when you put your chocks in, you'll get automatically moved into position on the catapult. Then power up, full flaps, etc. When you're ready, release chocks and the catapult will fire you off the end of the deck. Don't do any control movements til you're clear of the deck. Then raise gear, flaps, etc.

- AI aircraft, instead of taxiing into position for a rolling launch or catapult launch, now just "pop" into proper position for takeoff. This greatly simplified the coding, improved the performance, and fixed some lingering problems with AI running into the ship's island or off the edge of the deck, particularly in rough weather. Maybe a little less realistic...but that's the way it is.

- On carriers with two catapults, the AI only use the one on the left. Human players can use either one.

- There are several parameters that the mission builder can add to the [Mods] section of the mission (.mis) file that affect the use of the catapult. By default the catapult will work on all ships for both human players and AI. You only need to add any of these parameters if want to change the default behavior. NOTE: If you make any changes to the mission file in the Full Mission Builder after adding these parameters, they will disappear and will have to be re-entered:

CatapultAllow - Setting this to 0 disables the catapults on all ships for human players and AI. Default is 1.

CatapultAllowAI - Setting this to 0 causes the AI to use rolling takeoff on all ships. Human players can still use catapult. Default is 1. (This parameter can also be put in the [Mods] section of your conf.ini to set the default behavior of the AI.)

CatapultBoost - In dogfight and single-player missions the mission builder can set the parameter 'CatapultBoost 1' . This will increase the power of the catapult by approximately 30 knots. This gives you roughly the same results as you would have in a carrier moving at 30 knots in a co-op mission. Default is 1. When you need steam catapult powerful pushing, you have to set CatapultBoost 1 or Default.

CatapultPower - Catapult Boost power for Props, default =19. Need CatapultBoost=1.
CatapultPowerJets - Catapult Boost power for Jets after mission year 1945, default =25. Need CatapultBoost=1.
- Catapult power increasing at each year
MOD default parameters are changed by mission's year:
For pre-1945 CatapultPower=10 (CatapultPowerJets ignored and use CatapultPower_
At 1946-1950 CatapultPower=13 , CatapultPowerJets=19
At 1951-1954 CatapultPower=16 , CatapultPowerJets=22
At 1955-1958 CatapultPower=19 , CatapultPowerJets=25
At 1959- CatapultPower=22 , CatapultPowerJets=35
+PLUS 1953- adding push power variable with Fuel/Weapons Loadout

StandardDeckCVL - Setting this to 1 causes the catapult to be setup for the "standard deck" on the CVL's. Default is 0 (short deck). (This parameter can also be put in the [Mods] section of your conf.ini to set the default behavior of the AI and your aircraft.)

CatapultAI_CVE - Setting this to 0 causes the AI to use rolling takeoff on all CVE's instead of catapult. Default is 1. (This parameter can also be put in the [Mods] section of your conf.ini to set the default behavior of the AI.)

CatapultAI_CVL - Setting this to 0 causes the AI to use rolling takeoff on all CVL's instead of catapult. Default is 1. (This parameter can also be put in the [Mods] section of your conf.ini to set the default behavior of the AI.)

CatapultAI_EssexClass - Setting this to 0 causes the AI to use rolling takeoff on Essex and Intrepid instead of catapult. Default is 1. (This parameter can also be put in the [Mods] section of your conf.ini to set the default behavior of the AI.)

CatapultAI_Illustrious - Setting this to 0 causes the AI to use rolling takeoff on Illustrious instead of catapult. Default is 1. (This parameter can also be put in the [Mods] section of your conf.ini to set the default behavior of the AI.)

CatapultAI_GrafZep - Setting this to 0 causes the AI to use rolling takeoff on the Graf Zeppelin instead of catapult. Default is 1. (This parameter can also be put in the [Mods] section of your conf.ini to set the default behavior of the AI.)

CatapultAI_ClemenceauClass - Setting this to 0 causes the AI to use rolling takeoff on the Clemenceau Class instead of catapult. Default is 1. (This parameter can also be put in the [Mods] section of your conf.ini to set the default behavior of the AI.)

NoNavLightsAI - Setting this to 1 will cause the navlights on AI aircraft to stay off at night. Default is 0. (This parameter can also be put in the [Mods] section of your conf.ini to set the default behavior of the AI.).

FastLaunchAI - Setting this to 1 will cause the AI to move into launch position with their wings already unfolded. Default is 0. In co-op missions, players other than the host will still see the AI unfold their wings AFTER they've moved into position, and they may takeoff before their wings are completely unfolded. The wings will finish unfolding as they launch, and it will not affect the flight characteristics. The FastLaunchAI parameter is useful if you just want to get the AI off the deck as quickly as possible. For most aircraft it will not save much time, but for aircraft that unfold their wings very slowly (ex. F4U, B5N, B6N) it can save several seconds per aircraft launched. (This parameter can also be put in the [Mods] section of your conf.ini to set the default behavior of the AI.).

Sample [Mods] section in mission (.mis) file:

```
[Mods]
  CatapultAI_EssexClass 0
  NoNavLightsAI 1
  FastLaunchAI 1
```

Some AI-related parameters, as described above, can be put into the host's conf.ini instead of the mission file. The format in the conf.ini is slightly different. The parameter name and value are separated by an equal (=) sign instead of one or more spaces.

Sample [Mods] section in conf.ini file:

```
[Mods]
  CatapultAI_EssexClass=0
  NoNavLightsAI=1
  WingsFoldedAI=1
```

Just for Modders

i.e. How you can add these features to your new aircraft. Section will be updated as features are finished and new one are added.

How to add fuel dumping and other advanced features to your aircraft by SAS~Anto

I've designed the code to be both flexible and simple at the same time. Originally I believe this code was done by Flakiten but I have expanded it considerably. Using the new interface TypeFuelDump, you can define the following:

- Fuel Reserve: This is the amount of fuel that can't be drained. Essentially, take the total fuel and subtract the volume of fuel from the tank(s) you want to be drained (e.g. wing tip tanks). When you hit this value, fuel will stop draining out. Unfortunately you can't select which tank to drain (possible future feature) but with a bit of research, you can design it so that the most likely tanks will be emptied (whether it be wing tip tanks or in case of F-86s, either wings or main rear tank)

- Flow Rate: This is the rate (in litres per second) at which fuel is drained. Self-explanatory.

- In future, it may be expanded to include multiple tank definitions but for now kept it simple.

Here is an example of its implementation to the start of an aircraft class file:

```
package com.maddox.il2.objects.air;
public class Test_Plane extends Scheme1
  implements TypeFuelDump, TypeZBReceiver
{
  public static float FlowRate = 8.5F; //Flow rate in litres per second
  public static float FuelReserve = 1628F;//Minimum amount of total fuel that can't be drained - "Reserve Tank"
  public Test_Plane()
  {
  }

   //This method is called by AircraftState to read the flow rate defined here in the aircraft class
  public float getFlowRate(){
          return FlowRate;
  }

  //This method is called by AircraftState to read the fuel reserve defined here in the aircraft class
  public float getFuelReserve(){
          return FuelReserve;
  }
}
```

For the rest of the new interfaces (excluding TypeRadar), it is pretty straight forward. Implement the new interface and you will gain the new features. A few notes about some of them:

- For TypeRadarGunsight, it will add an empty method, which can be populated with code from the F-86s in 1956 project. Once populated and variables added, you need to call the method from Update(f). Then gyro gunsights will radar range.

- For TypeSupersonic, it also has a few methods attached. Again please have a look at the F-86s to populate the methods and add variables. In order to get supersonic effects, you need to call the supersonic method (should be identical to that in the F-86).

- TypeTankerDrogue and TypeTankerBoom will allow a tanker to dock with selected aircraft BUT alone, they will not work. You will need to have TypeDockable implemented and set the tanker up as the "mothership". See the Skyraider tanker or the KB-29 for more details

- TypeRadar is still WIP. It is modeled off the Acemaker interface and the controls will work in a similar manner. Watch this space ;)

Adding Drag Chutes to your aircraft by SAS~Anto

This is a little more complex in some manners. Currently we are hijacking the existing parachute class files to create this effect but are insert our own 3D (itself a clone of the parachutes used by pilots). At this moment, we are investigating making brand new chute types and making new chute classes (as extensions of existing chute class). So for now, watch this spot but this is the current implementation (on next page)

```java
package com.maddox.il2.objects.air;
public class Test_Plane extends Scheme1
{
    private boolean bHasDeployedDragChute;
    private Chute chute;
    private long removeChuteTimer;

public Test_Plane{}
  {
  }

public void update(float f)
    {
        super.update(f);
        //Deploys chute if key hit
        if(FM.CT.DragChuteControl > 0.0F && !bHasDeployedDragChute)
        {
            chute = new Chute(this);
            chute.setMesh("3do/plane/ChuteMiG21/mono.sim");
            chute.collide(true);
            chute.mesh().setScale(1F);
            chute.pos.setRel(new Point3d(-5.0D, 0.0D, 0.6D), new Orient(0.0F, 90F, 0.0F));
            bHasDeployedDragChute = true;
        }
        //Removes chute if overspeed, taxing slowly or released
        else if (bHasDeployedDragChute && FM.CT.bHasDragChuteControl){
            if(((FM.CT.DragChuteControl == 1.0F && (FM.getSpeedKMH() > 600F)) ||
FM.CT.DragChuteControl < 1.0F))
            {
                if(chute != null)
                {
                    chute.tangleChute(this);
                    chute.pos.setRel(new Point3d(-10D, 0.0D, 1D), new Orient(0.0F, 80F, 0.0F));
                }
                FM.CT.DragChuteControl = 0.0F;
                FM.CT.bHasDragChuteControl = false;
                FM.Sq.dragChuteCx = 0F;
                removeChuteTimer = Time.current() + 250L;
            }
            else if((FM.CT.DragChuteControl == 1.0F && (FM.getSpeedKMH() < 20F)))
            {
                if(chute != null)
                    chute.tangleChute(this);
                chute.pos.setRel(new Orient(0.0F, 100F, 0.0F));
                FM.CT.DragChuteControl = 0.0F;
                FM.CT.bHasDragChuteControl = false;
                FM.Sq.dragChuteCx = 0F;
                removeChuteTimer = Time.current() + 10000L;
            }
        }
        //Removes chute 3D after specified time after release
        if(removeChuteTimer > 0L && !FM.CT.bHasDragChuteControl)
        {
            if(Time.current() > (removeChuteTimer))
            {
                chute.destroy();
            }
        }
    }
}
```

Additional Flap Positions and Differential Brakes by Kumpel and Gerd(TAK)

1) Flap positions

Modification to AircraftHotKeys.class for key codes 52 & 53, where original code was placed with a dedicated public method SetFlapsHotKeys(int dir, RealFlightModel FM). This method must be assigned/replaced in AircraftHotKeysAlt.class - this way there is no need to make further changes to the main class when customising flaps function.

Key controlled flap action remains in 3 stages, but custom angles and hud messages can be assigned as required. For the latter to show, there must be a series of new messages placed in the hud_msg.properties file.

2) Differential brakes

The original implementation of differential braking in the game emulates typical behaviour of WW2 braking system - the power of braking controlled by a singled control (Brakes key) with the direction controlled by rudder angle. For aircraft with tricycle (vs. "tail-draggers") the front wheel is actively steered following rudder.

This mod introduces 3 types of brakes selectable by a control variable DiffBrakesType (accessible as FM.CT.DiffBrakesType) based on added proportional control functions that are meant to be assigned to left and right toe brake functions of pedals.

Type 0 - is the original type of braking with the addition of pedal action combined into a single control (Brakes key or Brakes proportional control).

Type 1 - works as type 0 but the front wheel is no longer actively steered by rudder, but passively follows aircraft movements (i.e. like many early tail-draggers).

Type 2 - is a semi-differential braking based on the original method with left and right brakes (Wheel Brake Right/Left Pedals or Wheel Brake Right/Left keys) acting on the corresponding main wheels. This allows for a more realistic brake control if pedals are used. Without pedals present, action reverts to Type 0.

Type 3 - true differential braking, with left and right brakes acting on corresponding wheels and common control acting evenly on both wheels (like a parking brake), activation by Wheel Brake Right/Left Pedals or Wheel Brake RIght/Left keys, front wheel passive steering only. As seen on the MiG-15/17 family and in real life, commonly featured on many Commonwealth aircraft.

Type 4 - is the original type of braking with NO pedal action combined, activated with Brakes key or Wheel Brakes proportional control. Spitfire & Yak type brakes.

The control variable DiffBrakesType is added to Controls.class and used in Gear.class, with default value 0. Individual aircraft classes must set this variable to get access to types 1-3. Independent Left and Right Brake functions were added to AircraftHorKeys.class, with both HOTAS (proportional) control and dedicated keys.

Java Examples for Developers (from A20 class)

1) To add check in aircraft for presence of active or passive steering, add following (default is active).

```
  protected void moveRudder(float f)
  {
    if(FM.CT.getGear() > 0.98F) {
      if(FM.CT.DiffBrakesType == 0)
        hierMesh().chunkSetAngles("GearC33_D0", 0.0F, 36F * f, 0.0F);
    }
    super.moveRudder(f);
  }


public void moveSteering(float f)
  {
    if(FM.CT.DiffBrakesType > 0)
      hierMesh().chunkSetAngles("GearC33_D0", 0.0F, -f, 0.0F);
  }
```

2) To set true differential brakes (or other types as you see fit)

```
//Added inside onAircraftLoaded. Note that the system print is optional but handy to have in beta stages

FM.CT.DiffBrakesType = 3;
System.out.println("*** Diff Brakes Set to Type: " + FM.CT.DiffBrakesType);
```

Credits

- Certificate's AI (HSFX5 version) with Anto's AI throttle fix and Nightfighter AI tweak. Additional AI tweaks by Recruit, Skipper and JG53_Valantine

- Carrier & Catapult Mod by Fireball, Benitomuso and western0221

- GATTACK mod by CY6, updated by Anto

- Ground Attack Orders mod by CY6

- Sniper Gunner fix by CY6, Riken, Lutz and Sputnikshock

- AI Overheating and Supercharger mod by Sani and Burn

- Zuti’s Friction mod

- Engine Mod by Sani and Anto

- Aircraft Hot Keys by Fireball, Zuti, Anto, Gerd, Kumpel and benitomuso

- PAT Air Show Smoke (extracted by F22-Raptor)

- Full Throttle Mod by Aed and Br!x

- New flap settings, differential brakes and steering by Kumpel and Gerd

- Compatible with realistic lights mod by Santobr

- Weapon control mod (inspired by Zloy_Peroshki), updated Fuel Dumping and Drag Chute by Anto

- Radar (still WIP and not yet functioning) by Anto and benitomuso

- SAS Superschool for testing

- Special thanks to SAS~Storebror for his patience and troubleshooting. Without him guiding me, this mod wouldn’t be here.