***ChinesePixel Presents:***

# VOLTAN ASSAULT TANK

***by Shen Hua, 2021 ©***

**This is a demonstration version of the Voltan Assault Tank Manual.**

**Buy the model to view the full version with XPressos described etc.**

## Introduction

This manual will help you familiarize yourself with the mechanics of the Voltan Assault Tank model. We will mostly focus on what each Xpresso is responsible for and what each controller does so you can know how to best operate this rig.

We will also briefly introduce the three material versions of the file so you can choose which one to use and see why we divided this model into 3 in the first place.

Once again, I would like to thank you for supporting my work, purchasing this model and I hope you will find it useful for production or education purposes.

I will be making more "Voltan" vehicles and troops in the near future so stay tuned for more if you're interested in this faction.

# Table of Contents

## A word on the Shader Network Types

This model is available as a package including 3 shader network types. All of them are based on Roughness/Metalness PBR:

- **PBR**, meaning the standard reflectance material layers PBR of Cinema 4D to represent basic PBR.
- **Nodal PBR** is also included. Here almost all materials for the model have been done with Nodal PBR system which is also compatible with Physical renderer of Cinema 4D.
- **Arnold PBR** is of course the Arnold's Roughness/Metalness PBR nodal system.

I have tried to make the 3 shader network types look as close to each other as I could. Of course, I recommend Arnold PBR for that is the intended look I was aiming for in this model and the other two are mostly included for compatibility reasons for those who don't use Arnold in C4D and still want to use this model.

*NOTE*: **Nodal PBR** requires more memory when rendering and renders much faster vs normal physical **PBR**.

Furthermore, keep in mind that both **PBR** and **Nodal PBR** version are compatible with same renderer and therefore if one material/part looks better under **Nodal PBR** or **PBR** you can mix them up to achieve the best result. And of course, **Arnold PBR** is separate and can't be mixed because it's a different renderer altogether.

There is a slight variation in polygon number between the 3 versions. Mostly between Arnold and the other two, because these use different light types and I added some proxy geometry for mesh lights in Arnold. Also, there's additional "arrow" mesh for rigged versions vs the ones that are just for static models.

## Tank Rig – Overview

**IMPORTANT: Remember to enable Expressions on all layers and link a terrain (page.8) to turn on the rig functionality.**

- Treads and wheels

The purpose of making this rig was to properly represent a modern SF tank with believable movement that won't require the use of dynamics but instead rely on splines, Xpresso, constraints and few very basic Python scripts threw in as Xpresso nodes.

Much detail was paid to the treads themselves. I decided not to use cloner (as that's the most popular way to do this, and I wanted to experiment with something that would give me more control) and instead used a chain model I got off Turbosquid as a base and combined it with Align to Spline tags and Xpresso.

As a result, we have a quite realistically behaving tread that reacts to the terrain you put the tank on (when properly linked, more on that later).

The tread is mostly built with two elements. The "track piece" which is aligned to the spline and spread around it with Xpresso and "joints", these are kind of hinges between these "track pieces" that connect them. The "joints" are connected with Aim Constraints.

I also made sure - using an Xpresso that has an "If then" behavior to drive some Aim Constraints - that the small driver wheels (Called "miniwheels" in the object list) will turn at a believable pace without having their geometry collide with the tread.

The result is very good all the way up to very sharp terrain where you might need to use special "shock absorber" and "suspension" controllers to manually adjust the tread further when keying the animation for even more precise control.

- Suspension

Suspension reacts to the terrain and also you can manually change the "hydraulics" position with controllers.

- Turret

Turret aims at the target "red ball" and you can change that it will either work as Inverse Kinematics (IK) or Forward Kinematics (FK) system. Turret Main Gun can either be "stabilized" and stay "on target" regardless of the pitch of the chassis or it can be "bouncy" as the tank is going over terrain. For aiming purposes it's best to use "Main Gun Direct Control" setting.

- Auxiliary Gun/Rocket Launchers

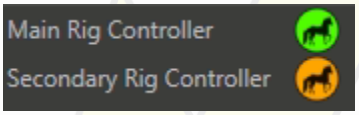
These weapons are mounted on the tank's turret. They can each either be set to follow the turret's rotation, follow their targets either in IK or FK fashion. There is also a "mass IK/FK" switch on the main tank controller to turn their IK into FK and vice versa.

- Fenders

At the back of the tank chassis there are two fenders hanging just above the treads. These are by default facing down regardless of the pitch of tank's chassis. This behavior can be disabled to get direct control over their rotation.

## Tank Rig – Mechanics and Controlling the Rig

- Main Rig Controllers (Tank Movement)




There are two main rig controllers. The difference between them is that one is a root controller and the other one is secondary. This means that the **Main Rig Controller** will rotate not just the chassis but also the IK targets of Aux Gun and Rocket Launchers. While FK targets for Aux Gun and Rocket Launchers will follow the turret's rotation. The same goes for the chassis movement. When you move the tank with the main controller, the IK targets will move along with it. While if you're using the **Secondary Rig Controller** for moving the tank chassis, the IK controllers will stay where they were and so the turret mounted weapons will keep facing them.

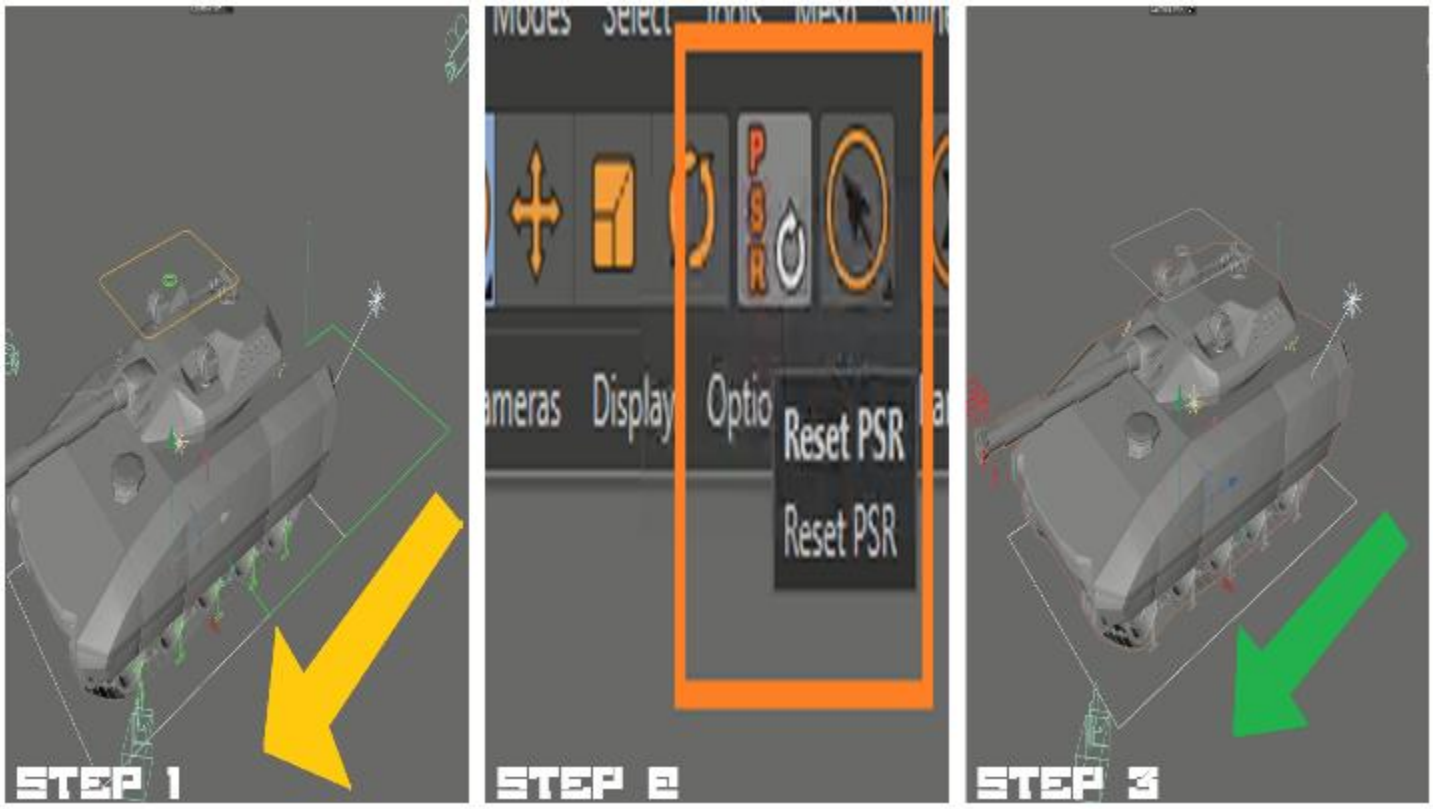
The Main Gun is an exception. By default, the Main Gun is aiming at the **Barrel Exit Controller**, the IK and FK switch (slider) of the Main Gun simply changes if you want the turret to rotate together with the chassis (FK) or you want it to keep facing the same direction while the chassis is rotating (IK).

Meanwhile if you move the chassis using either of the two main controllers, the turret's target (**Barrel Exit Controller**) will follow the movement while maintaining the general direction. This is to keep everything else synchronized and not to create any conflicts (contradictions) of how the weapons, turret and chassis should all move in relation to each other.

There is of course a way to also override this behavior and have direct control of Main Gun to make it "stay on target" even when the chassis is moving. All of that will be explained later in detail, for now let's see how the general combinations of IK and FK and different controllers work.

Naturally the best way to understand the various combinations is to try them out yourself.

*Keep in mind that once in a while you will probably want to use the **Main Rig Controller** <u>after</u> using the **Secondary Rig Controller** in your animation. That requires that you to first reset the PSR of both controllers and reposition the whole tank to where you have previously travelled and then start from there again in a sort of a "leap frog" movement. Otherwise the rotation of **Main Rig Controller** after using the movement of the **Secondary Rig Controller** will have the wrong point of origin as the **Secondary Rig Controller** moves away from the **Main Rig Controller**'s pivot. Using only the **Main Rig Controller** (Or directly the root model null on top of it) will not require any leap frogging but then again, you won't have as much control for the IK/FK of weapons.*




- Step 1 – Move/rotate the model using **Secondary Rig Controller (SRC)**.
- Step 2 – Make appropriate keyframes for your animation then click PRS for the **SRC.**
- Step 3 – Move the **Main Rig Controller (MRC)** to where your model was before you clicked PSR for the **SRC.**
- Step 4 – Now you can either continue using **MRC** or **SCR.** And repeat the "leapfrog" each time you used **SRC**.

- Combination Matrix

| CONTROLLER MANIPULATION | WEAPONS | | | | | | |
|---|---|---|---|---|---|---|---|
| | AuxGun IK | AuxGun FK | Rkt Launcher IK | Rkt Launcher FK | Barrel Exit IK (BEI) | Barrel Exit FK (BEF) | Main Gun Direct Control Enabled (BED) |
| MRC rotated (BEI) | FCR | SOF | FCR | SOF | SOI | - | - |
| MRC rotated (BEF) | FCR | FCR | FCR | FCR | - | FCR | - |
| MRC rotated (BED) | FCR | FCR | FCR | FCR | - | - | FCR |
| SRC rotated (BEI) | SOI | SOF | SOI | SOF | SOI | - | - |
| SRC rotated (BEF) | SOI | FCR | SOI | FCR | - | FCR | - |
| SRC rotated (BED) | SOI | SOF | SOI | SOF | - | - | SDT |
| MRC moved (BEI) | FCM | FCM | FCM | FCM | FCM | - | - |
| MRC moved (BEF) | FCM | FCM | FCM | FCM | - | FCM | - |
| MRC moved (BED) | FCM | FCM | FCM | FCM | - | - | FCM |
| SRC moved (BEI) | SOI | FCM | SOI | FCM | FCM | - | - |
| SRC moved (BEF) | SOI | FCM | SOI | FCM | - | FCM | - |
| SRC moved (BED) | SOI | FTR | SOI | FTR | - | - | SDT |
| BEI moved | SOI | FTR | SOI | FTR | SOI | - | - |
| BEF moved | SOI | FTR | SOI | FTR | - | SOF | - |
| BED moved | SOI | FTR | SOI | FTR | - | - | SDT |
| AGN = IK moved + BEI | SOI | - | U | U | U | - | - |
| AGN = IK moved + BEF | SOI | - | U | U | - | U | - |
| AGN = FK moved + BEI | - | SOF | U | U | U | - | - |
| AGN = FK moved + BEF | - | SOF | U | U | - | U | - |
| AGN = IK moved + BED | SOI | - | U | U | - | - | U |
| AGN = FK moved + BED | - | SOF | U | U | - | - | U |
| RKT = IK moved + BEI | U | U | SOI | - | U | - | - |
| RKT = IK moved + BEF | U | U | SOI | - | - | U | - |
| RKT = FK moved + BEI | U | U | - | SOF | U | - | - |
| RKT = FK moved + BEF | U | U | - | SOF | - | U | - |
| RKT = IK moved + BED | U | U | SOI | - | - | - | U |
| RKT = FK moved + BED | U | U | - | SOF | - | - | U |

--- LEGEND ---

Weapons/Controllers:

MRC = Main Rig Controller
SRC = Secondary Rig Controller
MDC = Main Gun Direct Controller
MGN - Main Gun Controller
AGN = Auxiliary Gun Controller
RKT = Rocket Launcher Controller
BEI = Barrel Exit/Main Gun set to IK
BEF = Barrel Exit/Main Gun set to FK
BED = Barrel Exit/Main Gun set to "Direct Target Control"

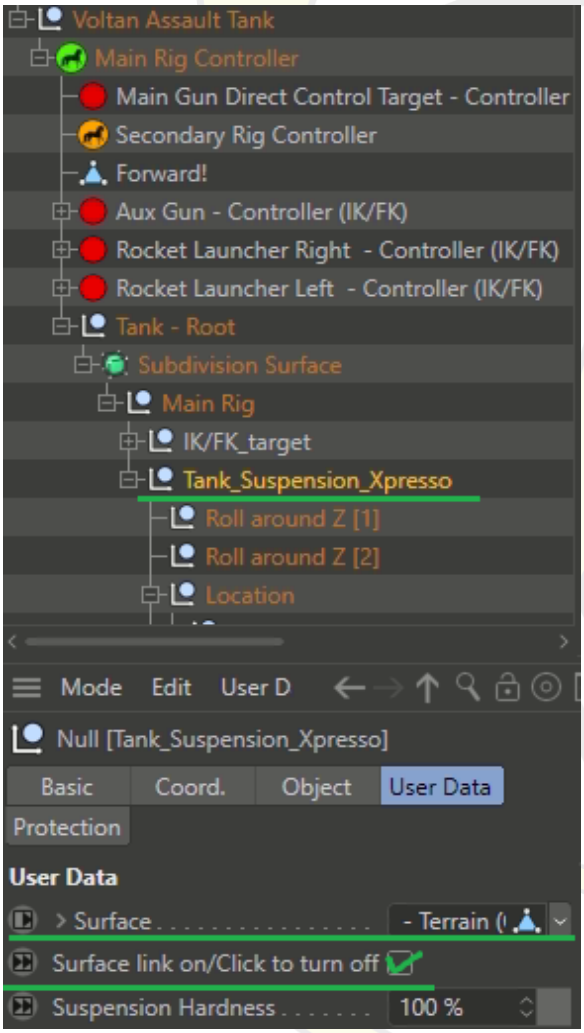
Behaviors:

SOI = Stays on IK Target
SOF = Stays on FK Target
SDT = Stays on Direct Target (For the Main Gun only)
FCR = Follows controller's rotation
FCM = Follows controller's movement
FTR = Follows turret's rotation
U = Unaffected
- = Does not apply

- Treads and Wheels. Linking terrain to model.

The wheels spin how they should in relation to the movement of the tread. The pace at which they spin is realistic meaning that to make the wheels spin faster you simply need to have them travel longer distance in a shorter time.

The vertical position of main wheels depends on the spline driving the tread. As your tread travels across obstacles the wheels will react to terrain, **as long as you have linked the terrain to the model**.



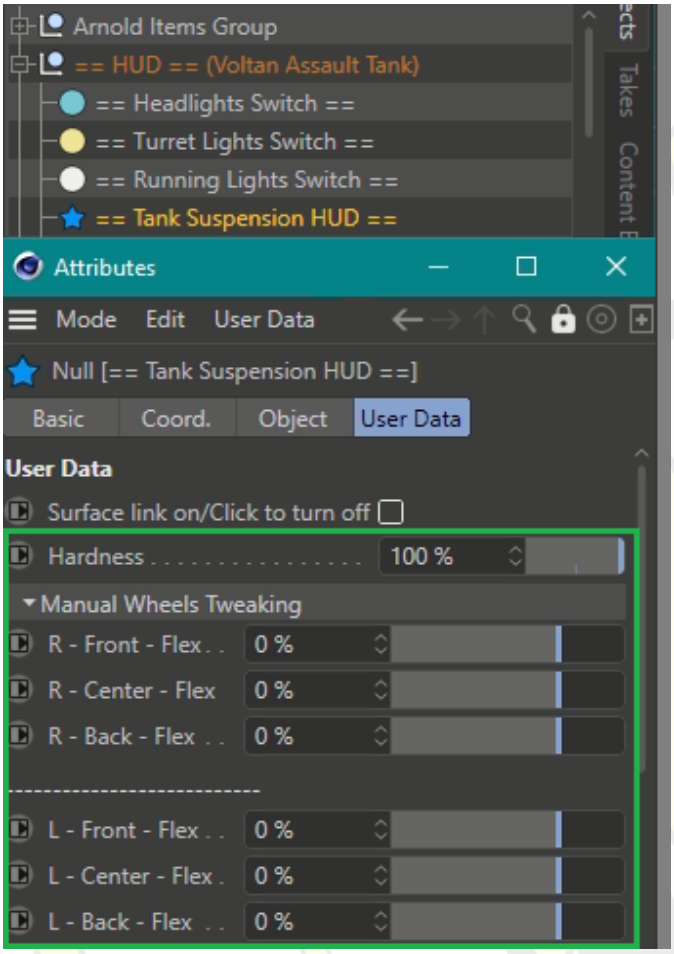

To link terrain to the model, you need to go to the rig null called **Tank_Suspension_Xpresso** and under its **User Data** there's a **Surface** dropdown selection where you can choose which object in the scene is the terrain you want to link to the tank.

*PRO TIP: if the terrain seems too rough and the tread is too low/high for your liking, what you can do is create a proxy (ghost) terrain object and link to it being offset a bit from the actual terrain displayed, so that the tank reacts to the proxy terrain while visibly it's driving on the terrain of size and position you decide to show to camera. Just keep the proxy terrain hidden during rendering. Slopes up to 22° have been tested.*

The settings for suspension and temporarily turning on and off the surface link are located at the **Tank Suspension HUD** null in the HUD section of the model. This is for the ease of control of the rig without a need to go to **Tank_Suspension_Xpresso** other than to link the terrain.

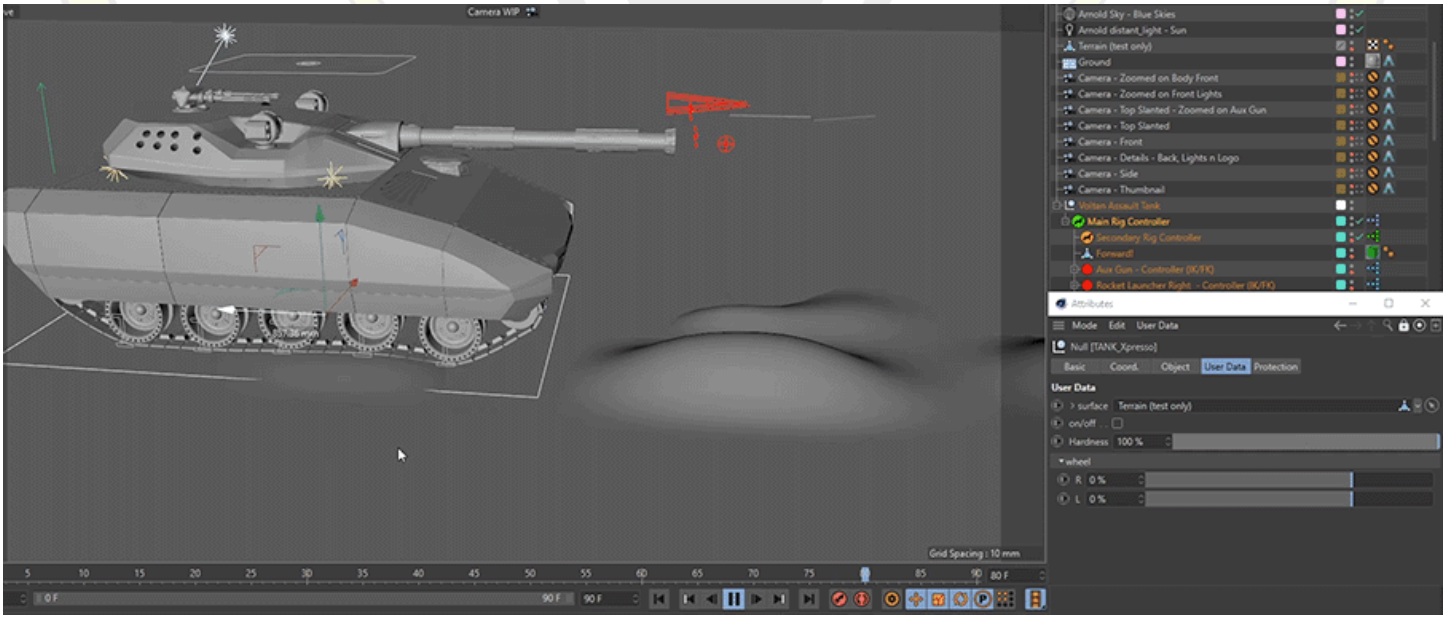
- Suspension

Suspension can also be adjusted at the same **Tank Suspension HUD** null. There is a slider for general **hardness** of suspension which controls how hard (suddenly) the suspension reacts to bumps and changes while making it softer gives it an effect of a slight delay and therefore feel much softer. Be aware that to see the difference you need to press play or be previewing the actual animation. It won't work while interacting with it in the viewport when it's paused.



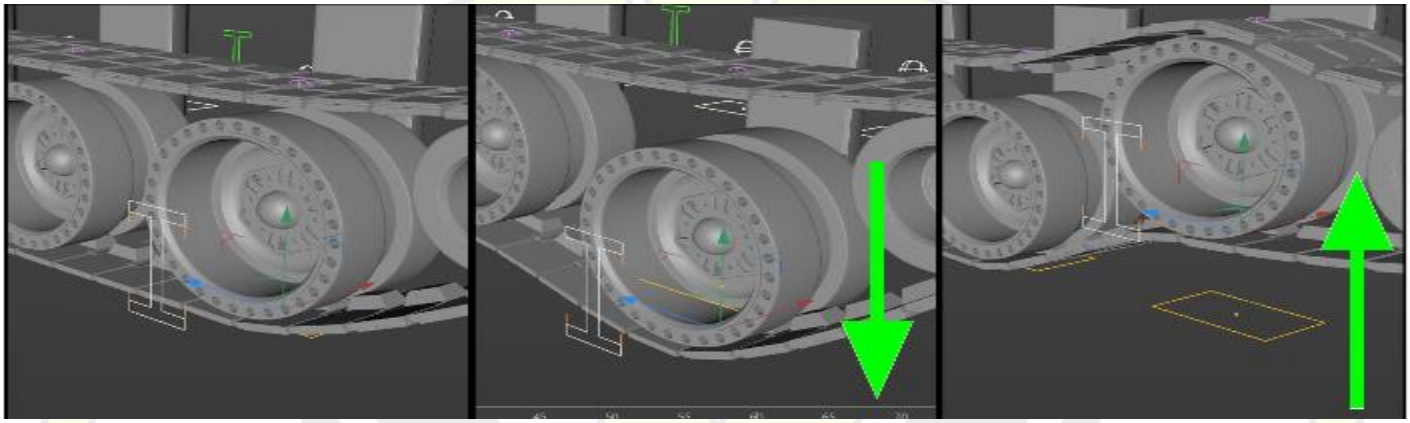

- Hardness – This controls how hard the suspension is and how quick it reacts when animation is played. 0% freezes the suspension. 1%-99% sets the desired delay factor. Setting the slider to 100% turns off the delay and therefore makes the suspension hard instead of soft.
- Wheel R - *** - Flex – Controls how tight the wheels are on the right side.
- Wheel L - *** - Flex – Controls how tight the wheels are on the left side.

And here's an example of how **hardness** affects the suspension (GIF animation only available in docx format of this readme because of PDF's limitations):

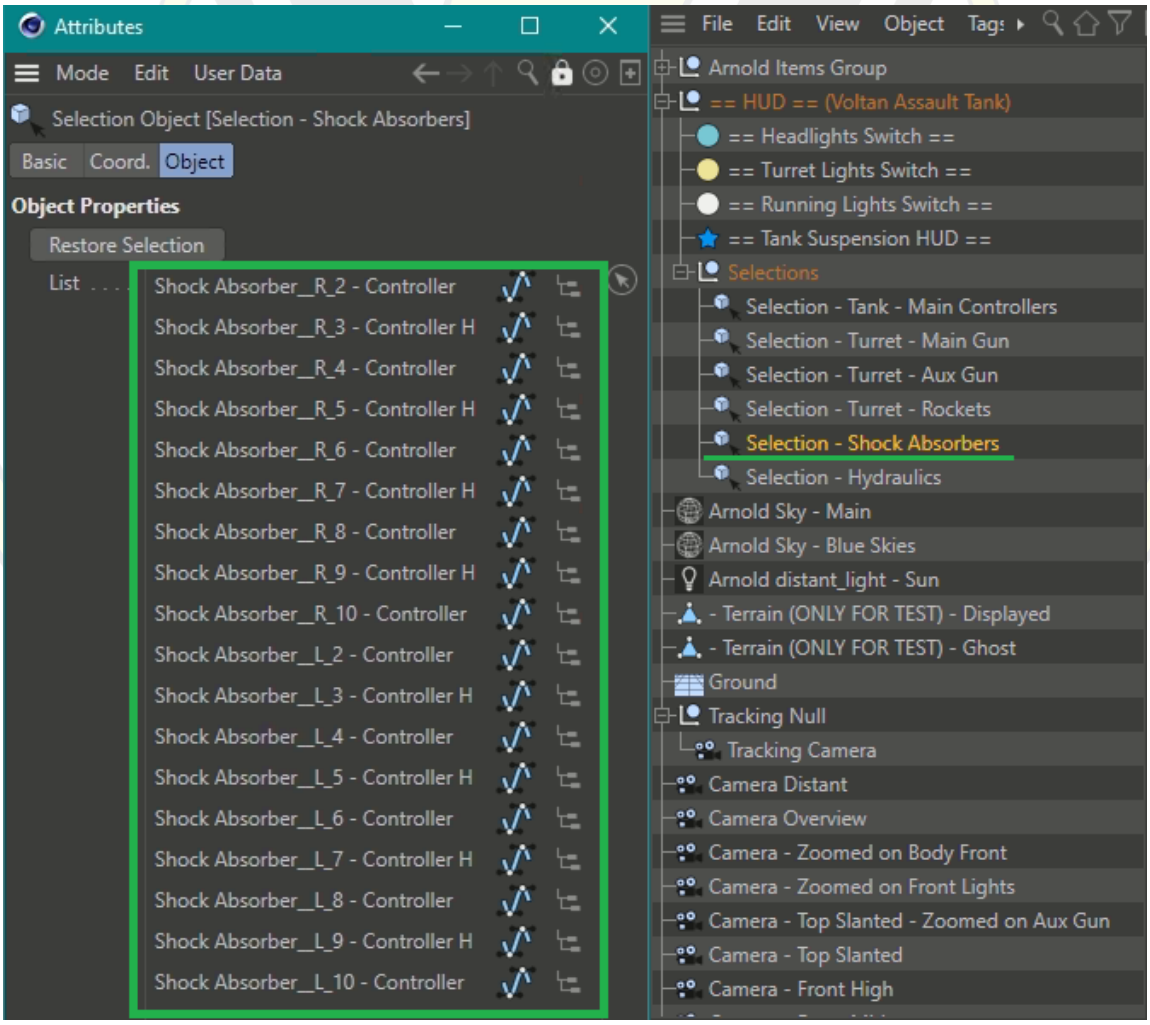


You may also manually adjust the height of each main wheel to make the animation even more precise. To do so you need to grab one of the **shock absorber** spline objects (There's a selection object in the scene that groups them all together). The controllers are shaped like I.

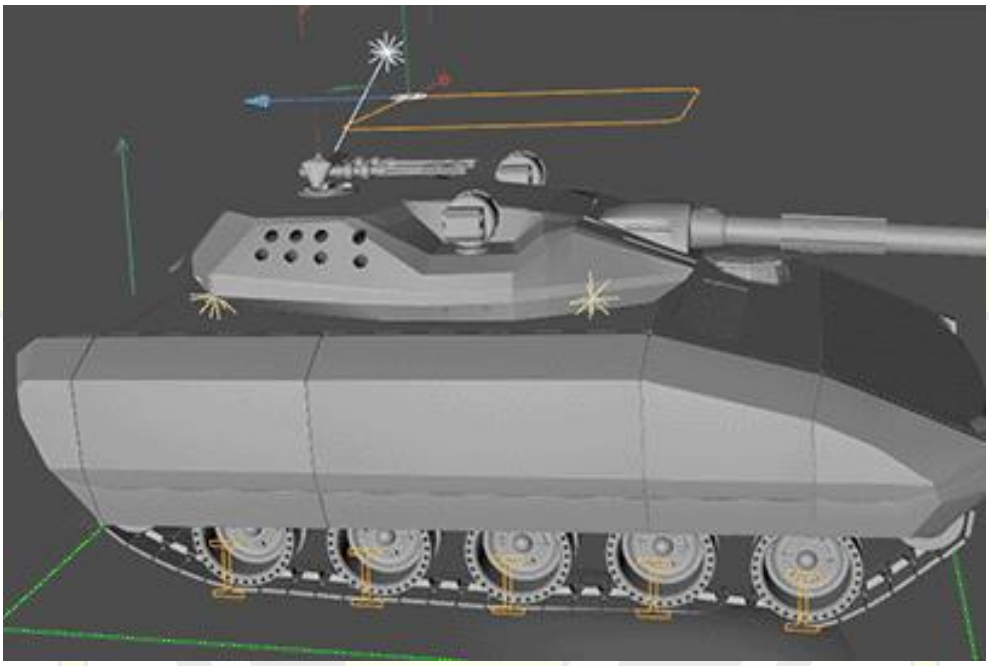
Here's how they work:



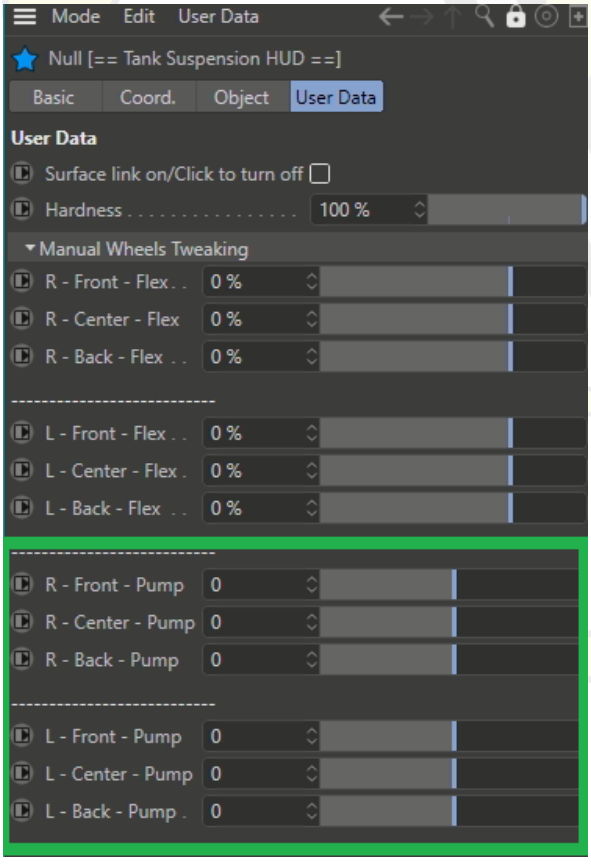
And here's where to find them:



You can also use a special **hydraulics** controller to change the tilt of the model. This controller is also found under one selection object called **Selection – Hydraulics**. You move the circle within the limits of a rectangle of the controlled to adjust the tilt to your liking.



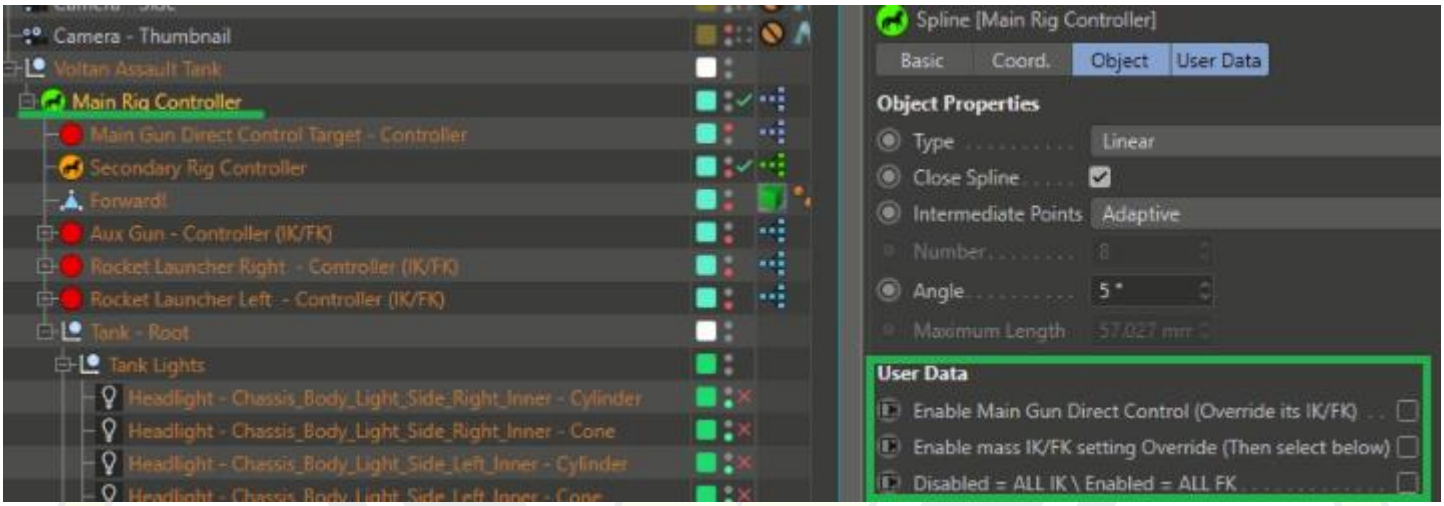
**Suspension HUD controls:**




With the **Tank Suspension HUD** controls, you can easily pump the wheels up and down, they're divided between left and right, front, center and back and it's quicker to use these controls than individual wheel and tread section with the **shock absorbers**.

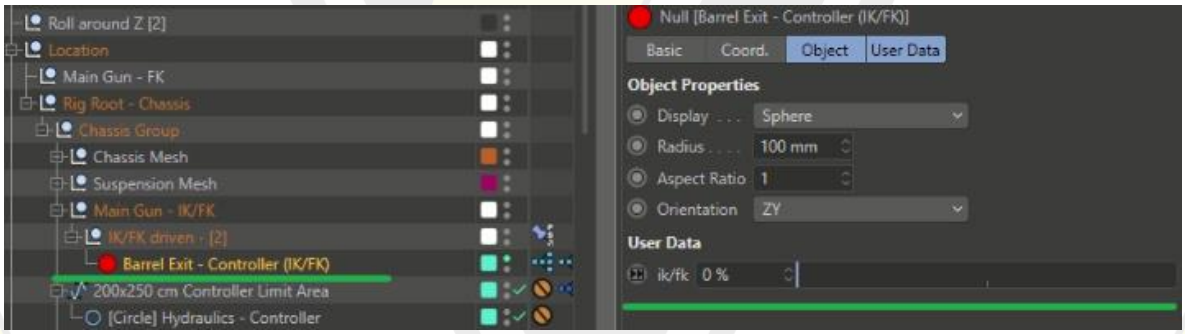
- Turret and the Main Gun.

You can change the heading of the turret and the pitch of the **Main Gun** using two different controllers. This depends on which setting is the turret **Main Gun** set to.

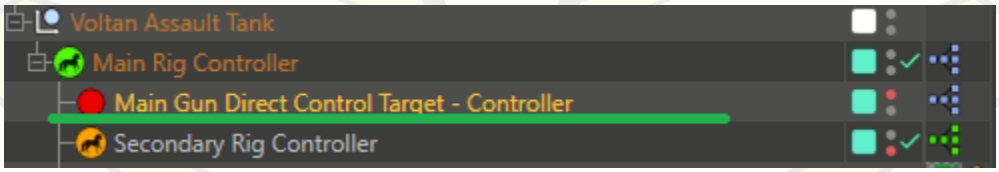



The settings are either set at User Data of **Main Rig Controller**. That's if you want to switch to **Main Gun Direct Control** or simply mass control IK/FK for all weapons including the **Main Gun**.

Or, alternatively you can just use the slider at User Data of **Barrel Exit** controller.



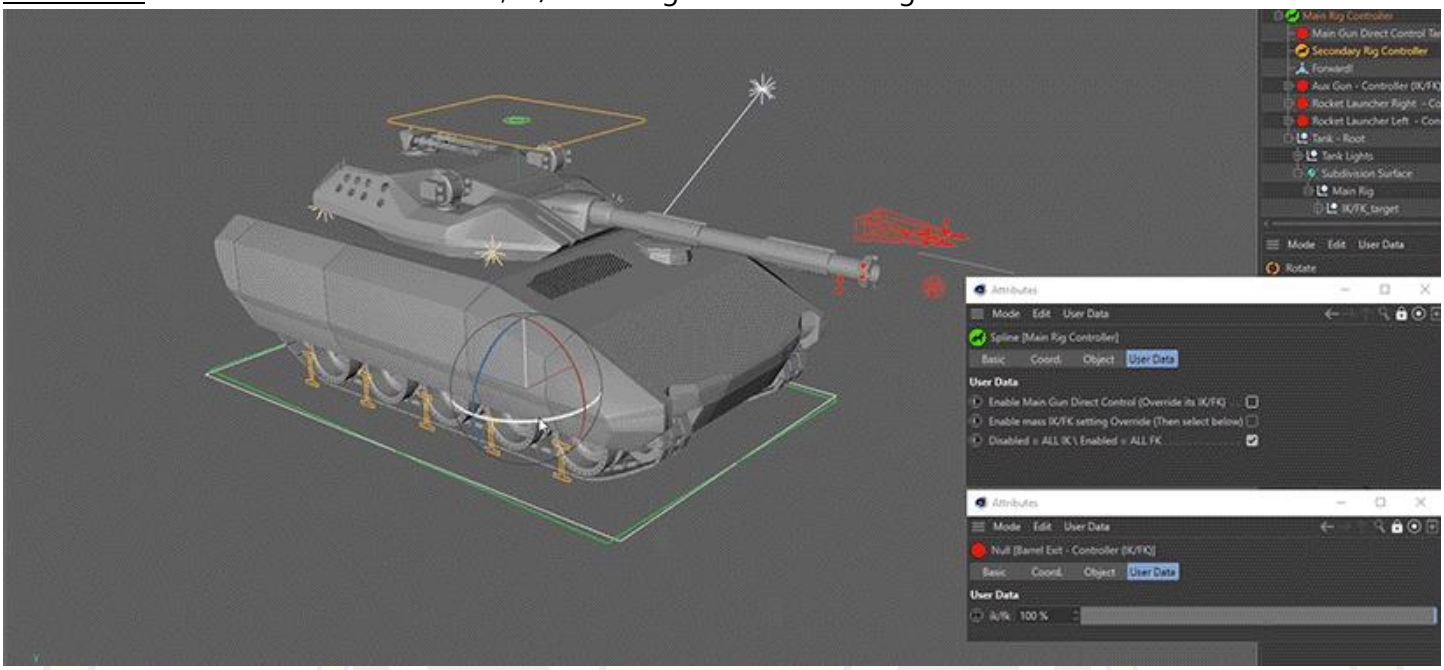

In case of IK and FK setting, you should use the **Barrel Exit**, meanwhile **Direct Control** overrides those and for that you should use the controller named **Main Gun Direct Control Target**.




The detailed **Combination Matrix** of these settings has been already discussed in previous pages [page 7]

IK and FK settings for the Turret/Main Gun differ mostly in that IK makes the turret stay on target when rotating the chassis using **Main Rig Controller** or **Secondary Rig Controller**, while FK follows their rotation. Both cases though will follow the movement of those controllers, therefore to have the most independent aiming of the **Main Gun** you should use the **Direct Control** setting.

Remember: when **Direct Control** is on, IK/FK setting of **Main Gun** is ignored.



This applies not only to heading but also the pitch. Which means that when moving through obstacles if you want the gun to stay on the target, use **Direct Control**, if you want to stay on the level of the target, use IK, and if you want the gun to move up and down together with the pitch of chassis, switch to FK.

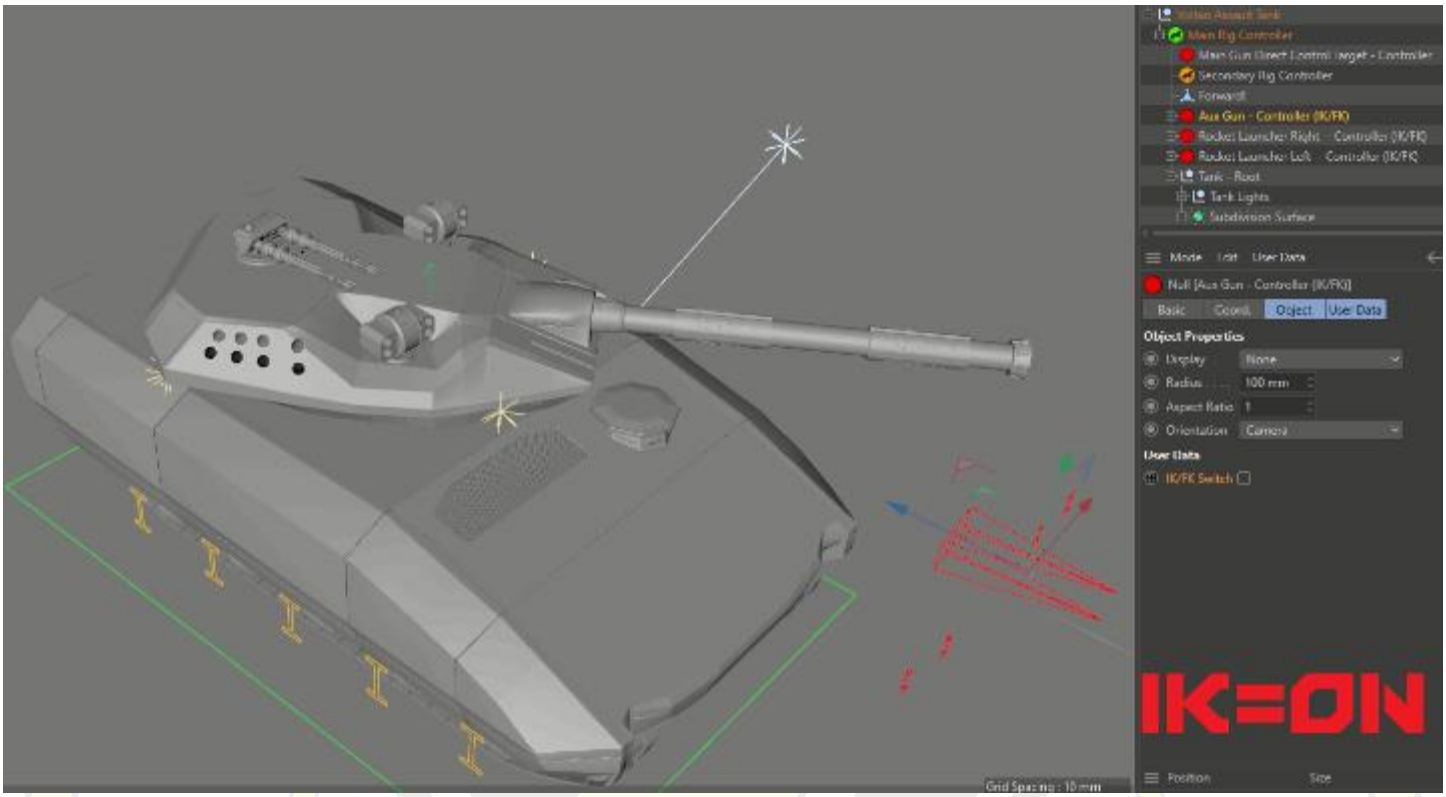
*Fluff Note: You might have been wondering what are the 8 holes on each side of the turret and on the back as well? Those are smoke grenade launchers used during the deployment of tank's troops.*

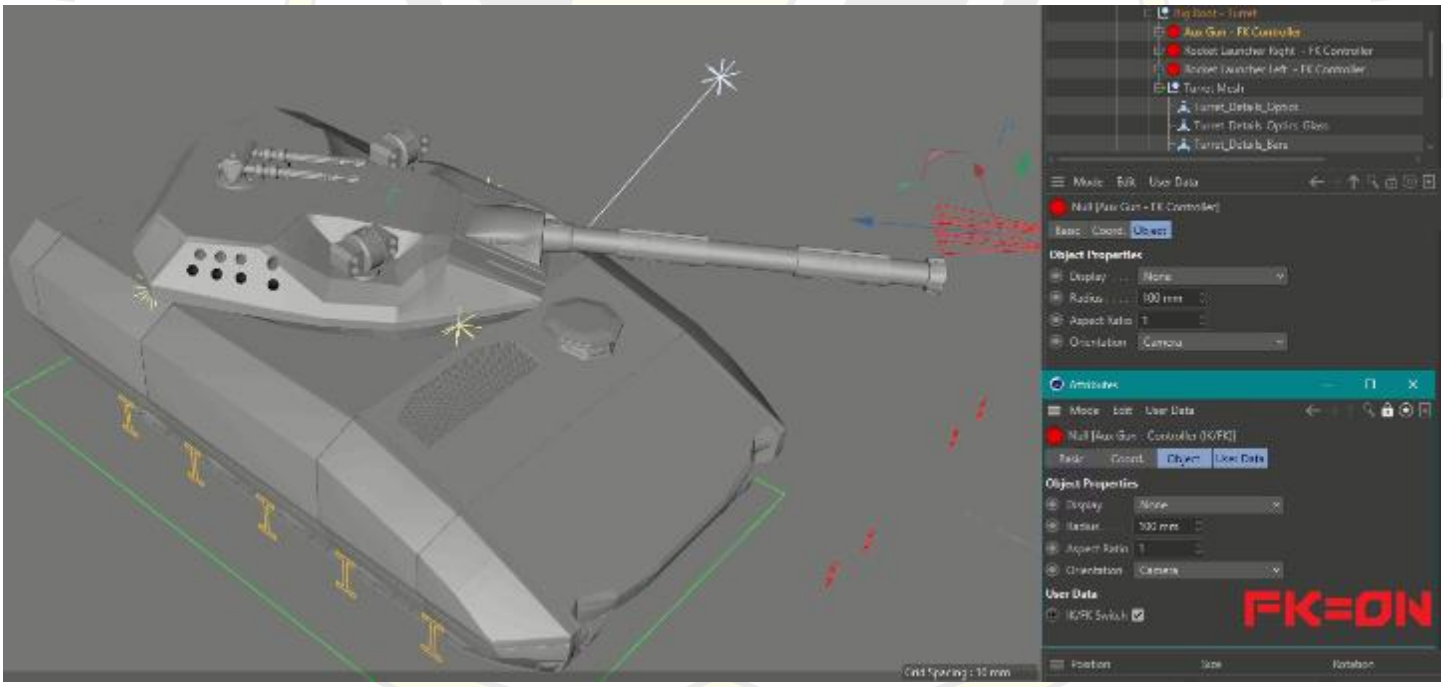
- Auxiliary Gun

Voltan Assault tank is also armed with a secondary gun, mostly for anti-aircraft defense but also for targeting infantry and lighter vehicles. This gun has twin-linked triple barrels.

There are two settings for this gun. IK and FK. Check the combination matrix to see how can you combine them with other settings.

When setting to IK, AuxGun will follow its IK target represented by the **Aux Gun – Controller (IK/FK)** and if you set it to FK then it will follow the **Aux Gun – Controller FK** as the target. FK setting also means that both the AuxGun and its target are children of the Turret while IK means that the target is higher in hierarchy and therefore more independent.


IK=ON


FK=ON

- Rocket Launchers

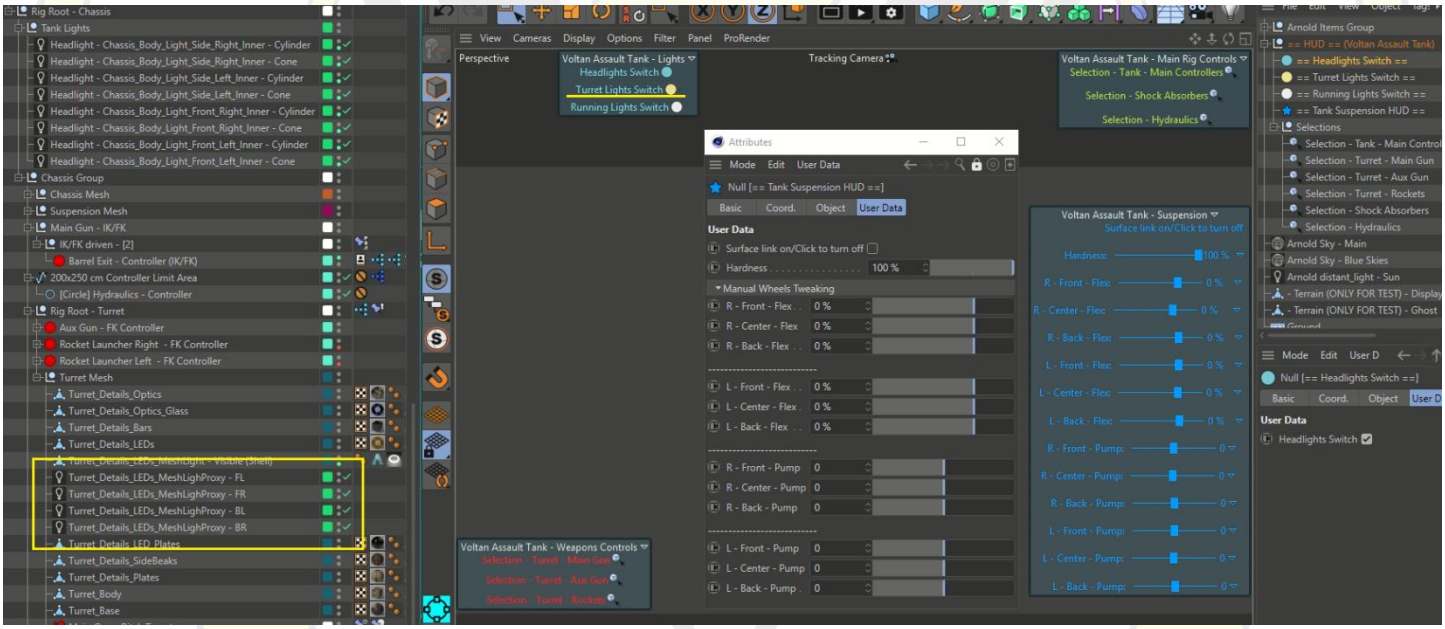
This tank is also armed with two batteries of **Rocket Launchers**. They are able to shoot rockets or mortar charges as an artillery weapon.

The principle behind settings and using **Rocket Launchers** is same as for the **Auxiliary Gun.** The only difference being that the controllers would be named **Rocket Launcher Right/Left (IK/FK)** accordingly.

## HUD and Lights

The HUD contains quick access to selection objects. Use it to select the rig controllers like **Main Controllers, Shock Absorbers, Hydraulics, Suspension.** HUD also lets you quickly select controllers related to the weapons: **Main Gun, Aux Gun, Rockets.** Using HUD, you can also turn the tank lights on and off.

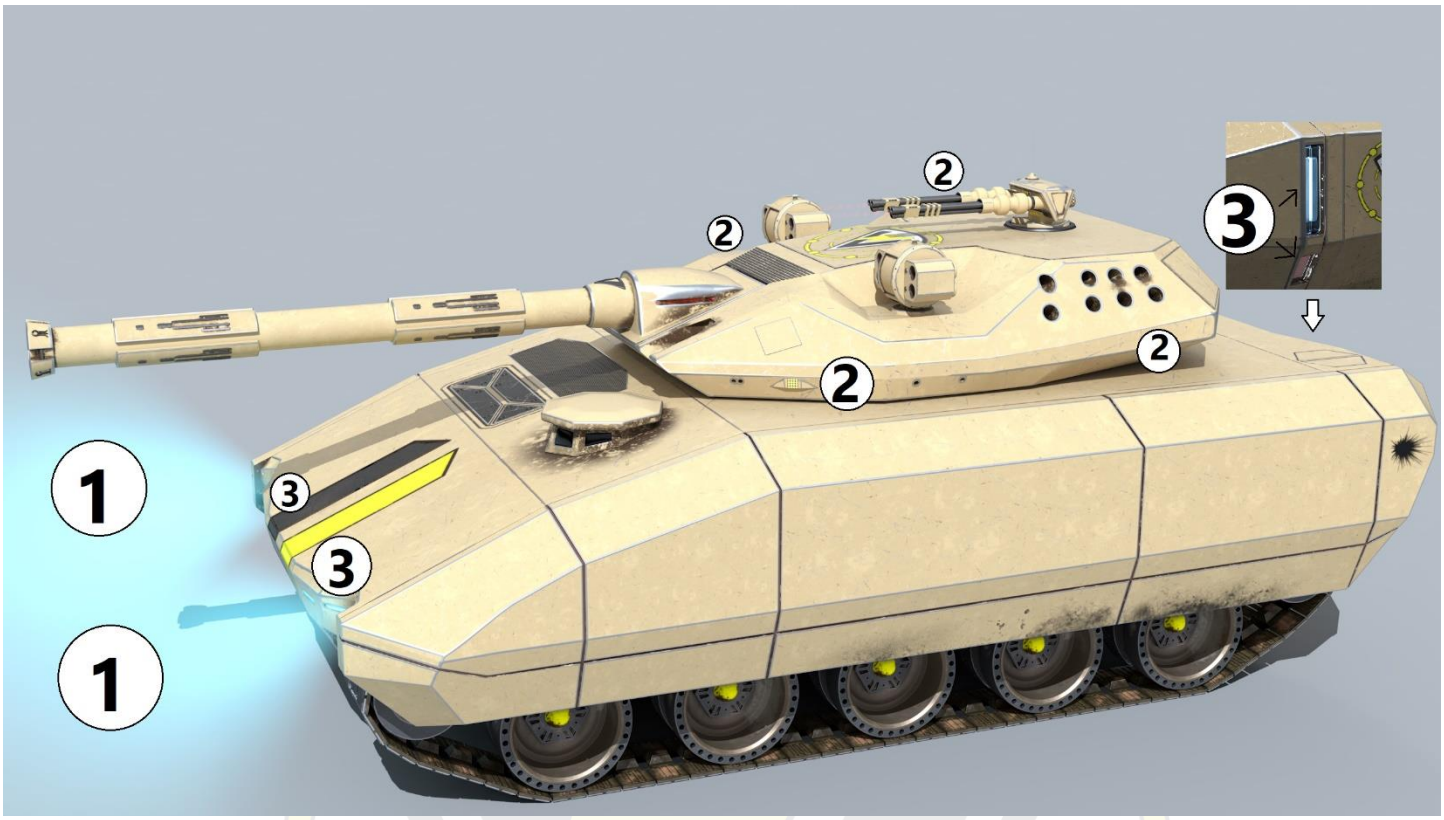
- HUD




- Lights – Overview

Lights are divided into three main groups:

1. Headlights. These are high beams from the front lights. Off by default.
2. Turret lights. Yellow lights around the turret. On by default.
3. Running lights. These are the back bluish and red and also front bluish lights. These are on by default.




- Lights – Arnold

Arnold version of this model lights are done this way:

1. Headlights are composed of **cone lights** to give the **volumetric light** beam (there's also an **atmosphere** in the scene to accommodate that.) and **cylinder lights** to fill out the light "box" more.
2. Turret Lights are a group of 4 **point lights** around the turret. They also involve a **mesh light** made of a **"shell" proxy geometry** that doesn't get subdivided to make sure there's no noise issue with the **mesh light**. That mesh light part is controlled by running lights controller.
3. Running Lights. Those are **mesh lights** divided in two parts. **The Light** part and **the Rays** part. The Light part is a **mesh light object** that's projected on the geometry of the light. The light is visible for this one but rays shining from it aren't. Then the other half is rays. That is a copy of geometry that the mesh light represents with a **mesh light tag**, that's not subdivided to avoid noise issues, and it doesn't have a visible light source, but the rays are active. This set up works the best from my experimenting with rendering these lights.

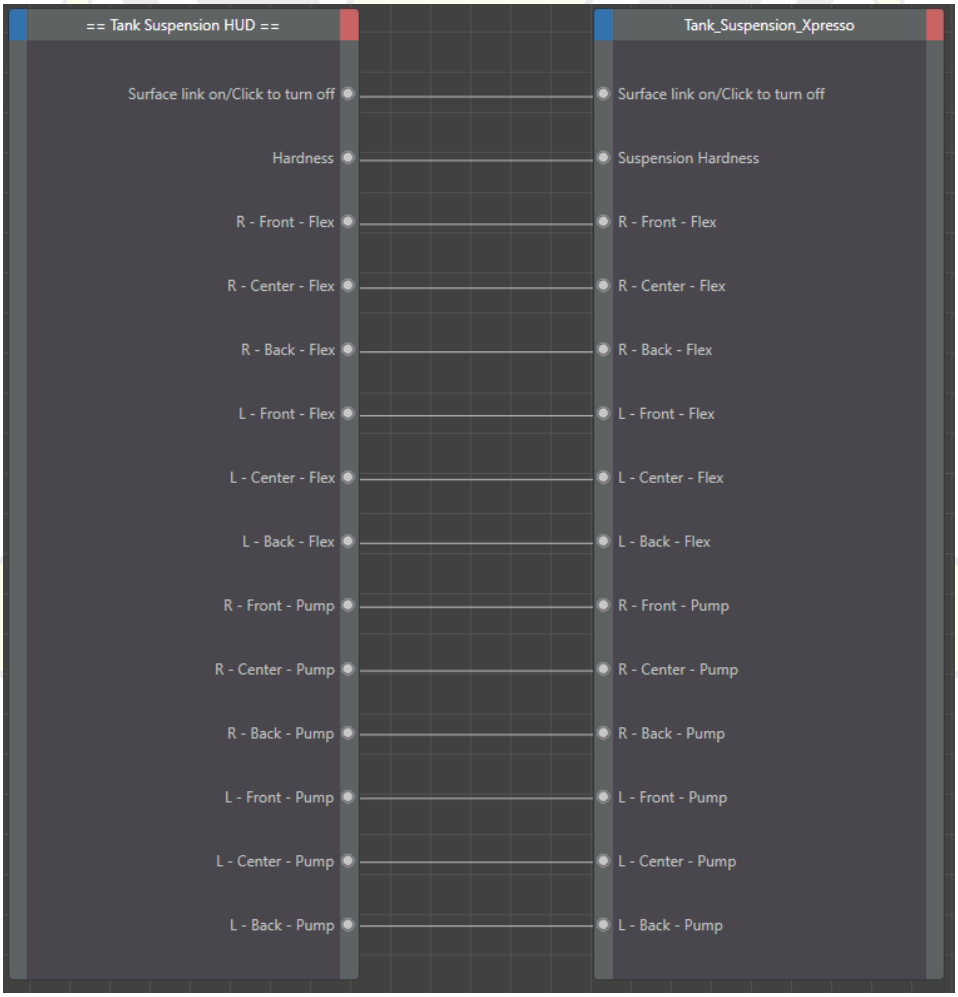
- Lights – PBR and Nodal PBR

For Non-Arnold version of this model the lights work this way:

1. Headlights are made with **spot lights** that have a basic volumetric look to them. (Pictured on page 16)
2. Turret Lights are a mixture of **omni lights** and a bit of **emission**.
3. Running lights are **area mesh lights**.

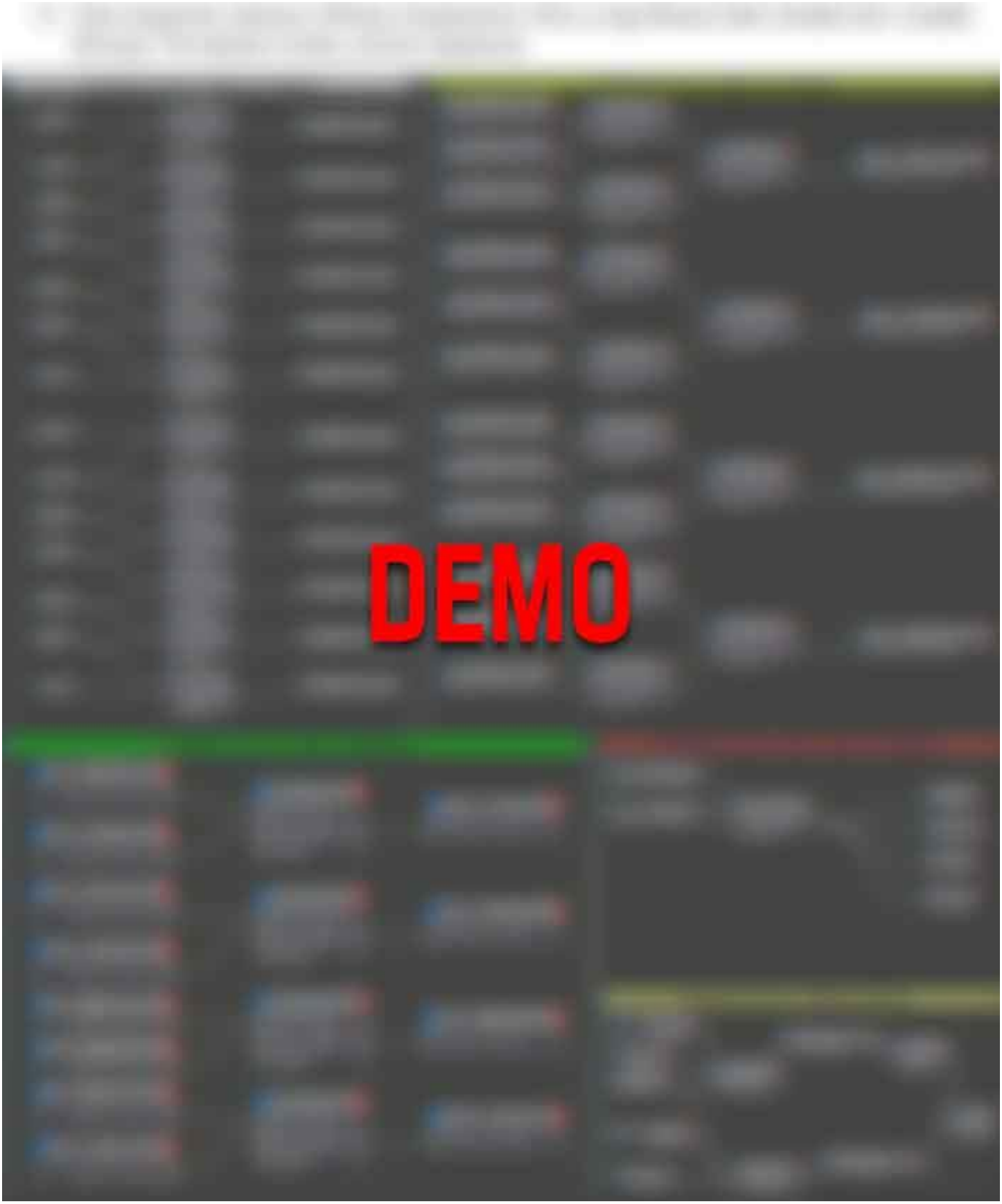
**XPressos (This section is available only in the full version of this manual)**

01. == Headlights Switch == | XPresso | Controls group #1 of lights.
02. == Turret Lights Switch == | XPresso | Controls turret lights.
03. == Running Lights Switch == | XPresso | Controls group #3 of lights.
04. == Tank Suspension HUD == | XPresso | This XPresso simply relays the control from **Tank_Suspension_Xpresso** null to the **HUD** null for your convenience.

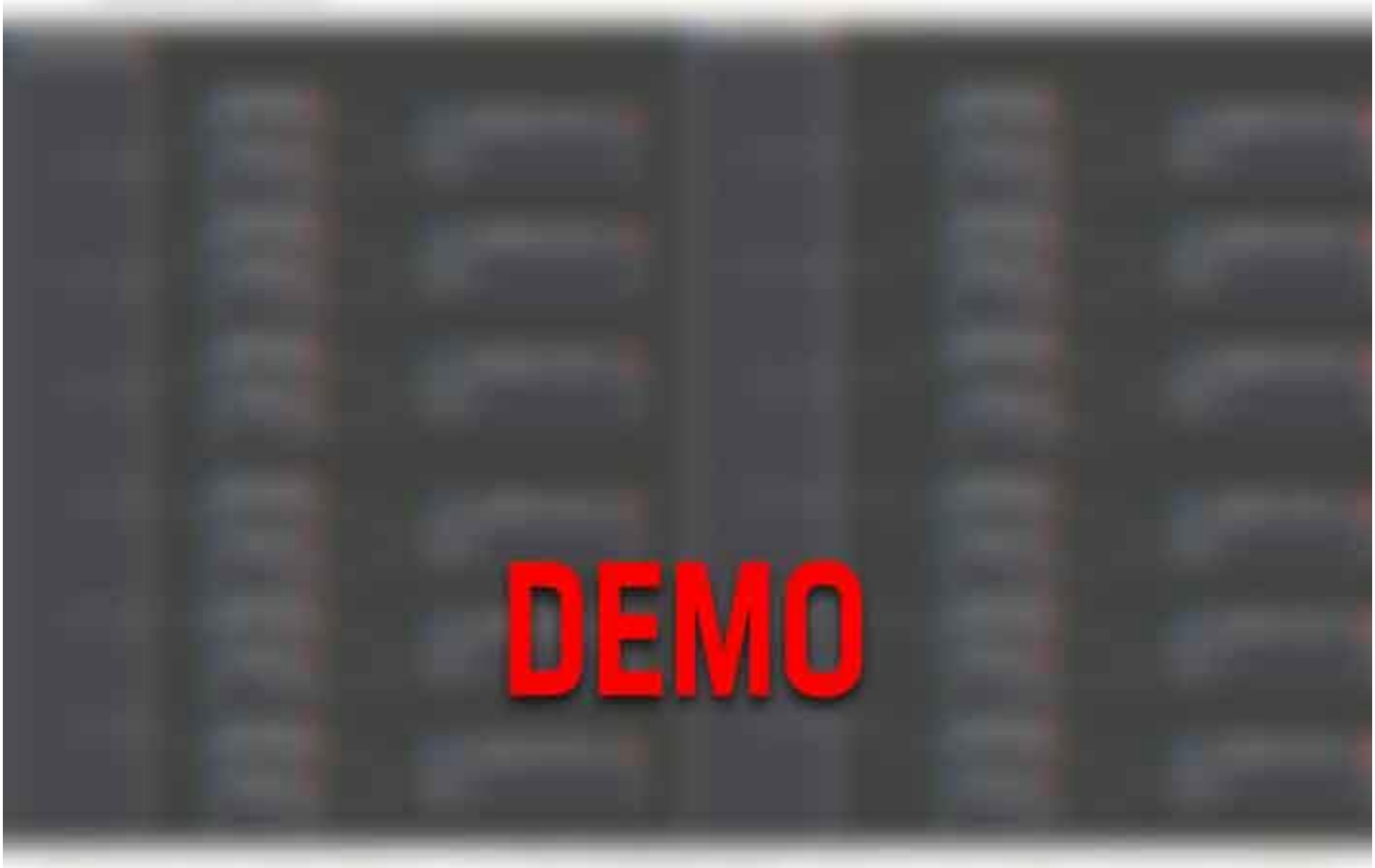



DEMO


DEMO

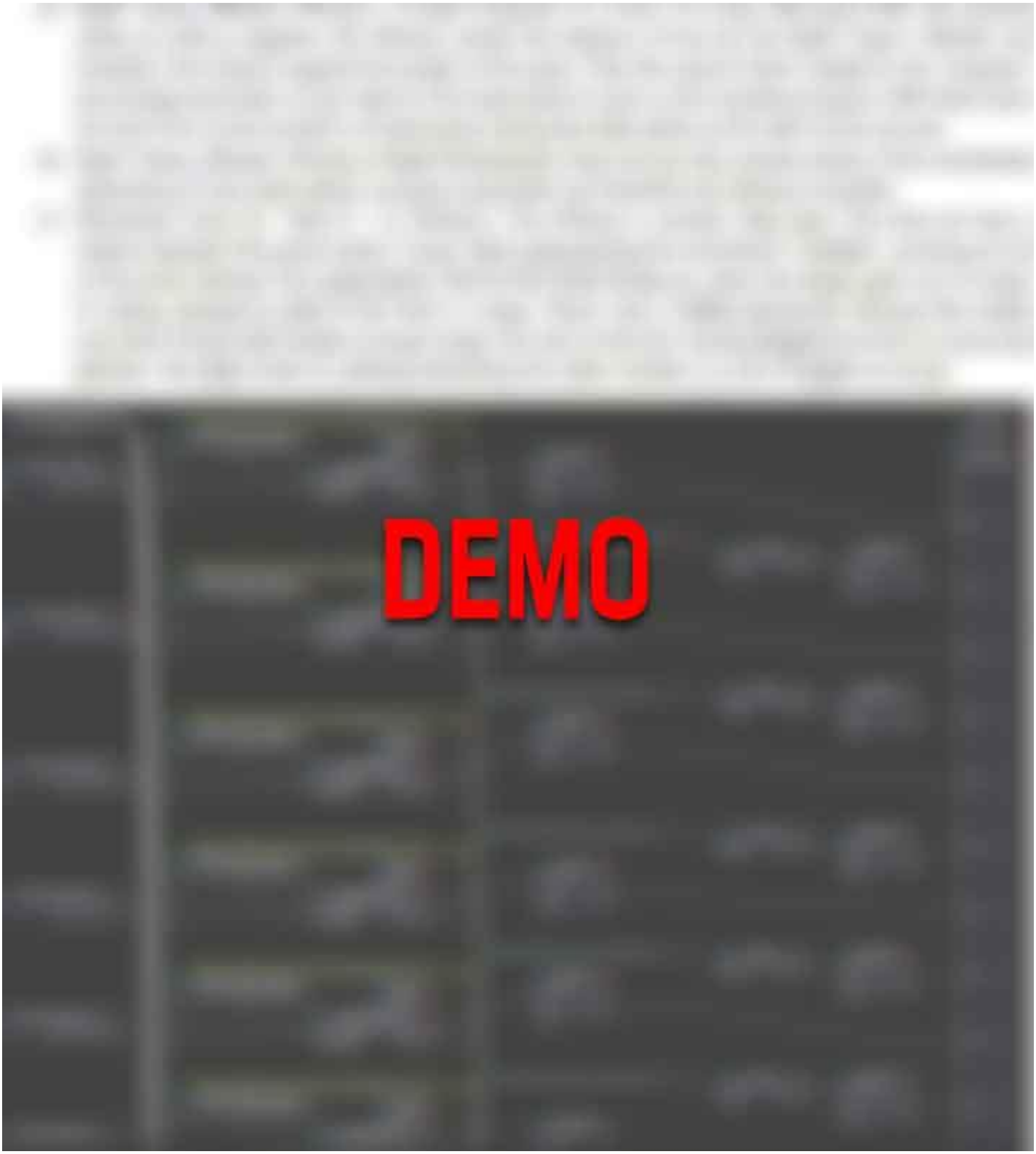

DEMO


DEMO


DEMO


DEMO


DEMO


DEMO


DEMO


DEMO


DEMO


DEMO


## Model Statistics

| Rested Nodal PBR Version | Rested PBR Version | Rested Arnold PBR Version |
|---|---|---|
| 1811614 Points | 1811614 Points | 1540846 Points |
| 1799356 Quad Polygons | 1799356 Quad Polygons | 1528428 Quad Polygons |
| 888 ( 898 ) Objects | 888 ( 898 ) Objects | 908 ( 929 ) Objects |
| 0 Controllers | 0 Controllers | 0 Controllers |
| 148 Materials | 150 Materials | 150 Materials |
| 115 Unique UV Maps | 115 Unique UV Maps | 114 Unique UV Maps |
| 50 XPressos | 50 XPressos | 50 XPressos |
| Textures are 4K and 8K | Textures are 4K and 8K | Textures are 4K and 8K |
| **Rigged Nodal PBR Version** | **Rigged PBR Version** | **Rigged Arnold PBR Version** |
| 1894668 ( 1916434 ) Points | 1894668 ( 1916434 ) Points | 1623900 ( 1645666 ) Points |
| 1881374 Quad Polygons | 1881374 Quad Polygons | 1610446 Quad Polygons |
| 1583 ( 1629 ) Objects | 1583 ( 1629 ) Objects | 1602 ( 1659 ) Objects |
| 30 Controllers | 30 Controllers | 30 Controllers |
| 150 Materials | 150 Materials | 152 Materials |
| 115 Unique UV Maps | 115 Unique UV Maps | 114 Unique UV Maps |
| 50 XPressos | 50 XPressos | 50 XPressos |
| Textures are 4K and 8K | Textures are 4K and 8K | Textures are 4K and 8K |