



Viz Artist 5.2

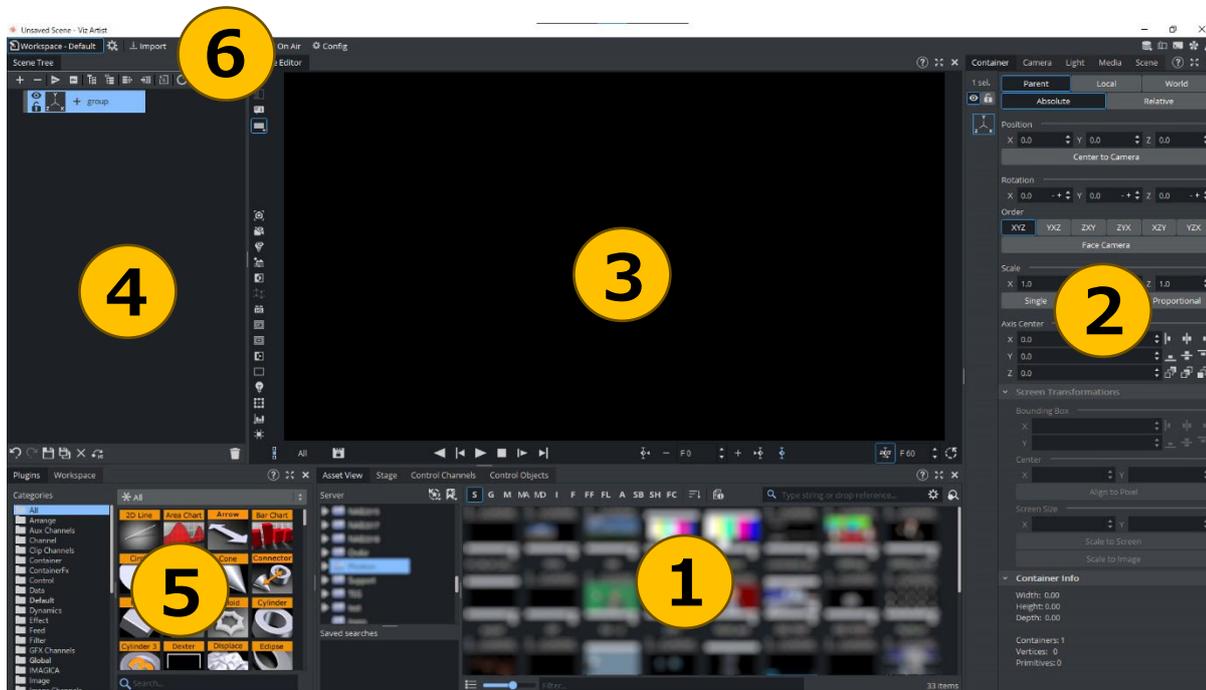
アドバンスドトレーニング Vol.1

Photron

v5.2.1 2024.10.29

Viz Artistユーザーインターフェース(概要)

Viz Artist を起動すると以下のような画面が表示されます。



メインウィンドウの上部には、メインメニューバーがありメニューバーの下には各表示エリアがあります。各表示エリアは、タブをドラッグすることで移動・ドッキング/ドッキング解除、またサイズ変更ができます。各エリアの詳細は実習の中で説明します。

① アセットビュー

VizGH に保存されたシーン、ジオメトリ、マテリアル、画像、フォントなどの各種アセットを表示します。

ツリー状に複数フォルダを作成し、アセットを整理することができます。

ツリー上のフォルダを選択すると右側にフォルダ内のアセット一覧が表示され、上図のボタンを押すことで種別表示になります。

- ステージエディタ

アニメーションのタイムラインを表示し、キーフレームの登録設定など、アニメーション作成を行うことができます

- コントロールパネル

外部制御ソフトから参照するコントロールチャンネル/コントロールオブジェクトを表示し登録することができます

② プロパティパネル

シーンエディタやシーンツリーで選択した「コンテナ」、「カメラ」、「シーンライト」、および「メディア」に関する各パラメータを表示し設定することができます。

また「シーン」のレンダリング種別指定やポスト処理の設定などを行います。

各パラメータの数値をクリックし左右にドラッグすることで数値を変更することができます。またパラメータの名前をダブルクリックすることでデフォルトの値になります。

③ シーンエディタ

選択されているカメラのレンダリングプレビューが表示されます。

カメラは左上の数字ボタンを押すことで切り替えることができます。

※Front～Bottom ボタンはカメラ切替えではなく、シーンの手前奥・左右・上下から見た平行投影を表示します。

④ シーンツリー

シーンツリーは、シーン内のすべてのコンテナの親子関係をつリー状に表示します。また各コンテナが持つプロパティをコンテナ上にアイコンで表示します。（アイコンをクリックすることでプロパティエリアに各種パラメータが表示されます）

不要なコンテナ/追加されたプラグイン等を削除する場合、マウス右クリックメニュー/Delete ボタンを押す。

またはゴミ箱アイコンにドラッグ & ドロップすることで削除することができます。

⑤ プラグイン

インストールされたプラグイン及びメディアアセットの一覧を表示します。

- ジオメトリプラグイン : 形状
- コンテナプラグイン : コンテナに機能を追加
- シーンプラグイン : シーンに機能を追加
- シェーダープラグイン : コンテナにシェーダ機能を追加
- メディアアセット : 外部入力映像や動画クリップ

⑥ メインメニュー

画面を各パネルに切り替えます。

- Workspace メイン制作画面
  にてワークスペースのプリセット切り替えや、保存などが可能です。
- Import 各種素材またはアーカイブファイルを選択し Viz Graphic Hub に読み込みます。
- Archive 指定したアセットをアーカイブファイルに出力します。
- Post ポストレンダリングパネルを開きます。
- On Air アプリケーションモードを制作環境からプレーンなレンダリングエンジンに切り替えます。
アプリケーションは、通信ポート（デフォルト：6100）を介して制御コマンドを待機するようになります。
- Config システムオプションと機能を変更するパネルを開きます。

最新のViz バージョン新機能紹介(抜粋)

■ Viz Artist 5.1 ベーシックマニュアル

https://www.photron.co.jp/service/videosolution/support/viz/technical_info/102503.html

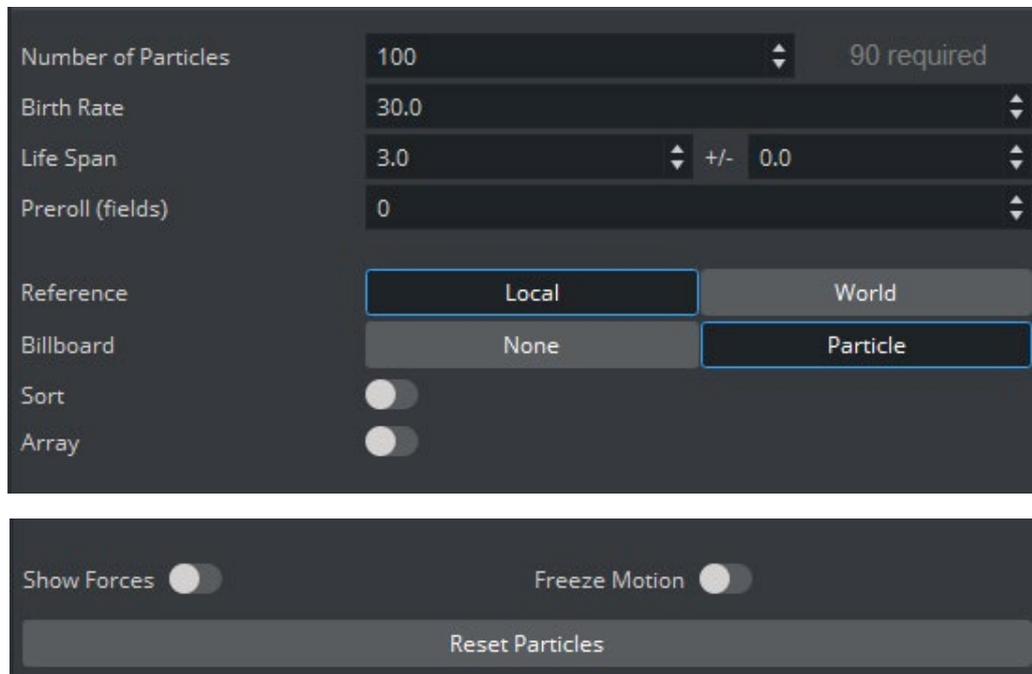
- Post Processing
- Global Illumination
- Instance
- VML Clip Player
- Texture Slot
- Sub Scene
- DVE / GFX
- Scene Overview



1. パーティクル (RFxSmoke) について



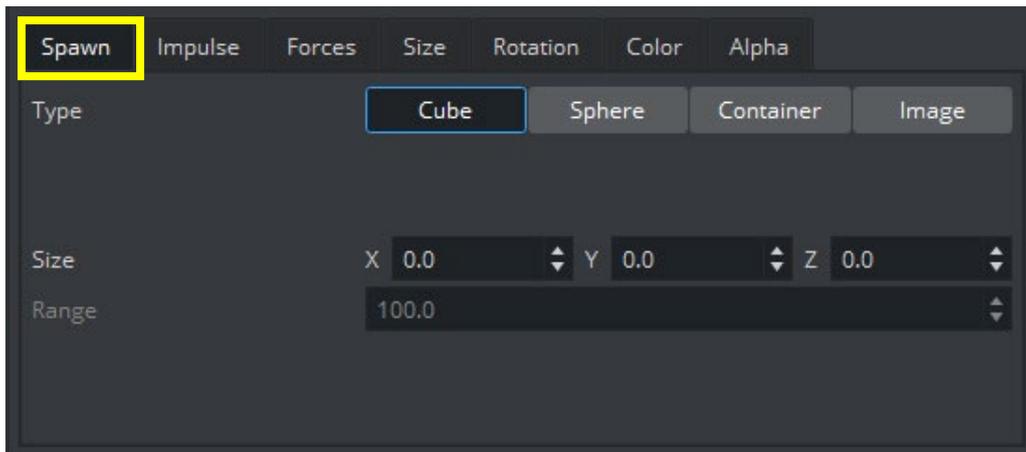
RFxSmoke プロパティ



Number of Particle	同時に表示される粒子の数を設定
Birth Rate	1 秒あたりのパーティクルの発生率を設定
Life Span	粒子の寿命を秒単位で設定 +/-で寿命のばらつきを設定 (%)
Preroll	オンエアになる前に計算するフィールドの量を設定
Reference	コンテナベース、もしくは空間ベースで粒子の座標を維持
Billboard	カメラに対して粒子の面を、常に正面の方向を維持
Sort	並べ替えを有効/無効にします
Array	配列の使用を有効/無効にします

Show Foces	一部のパラメータの設定を示すヘルプラインを表示
Freeze Motion	粒子のアニメーションを停止
Reset Particles	生成されているパーティクルを消去

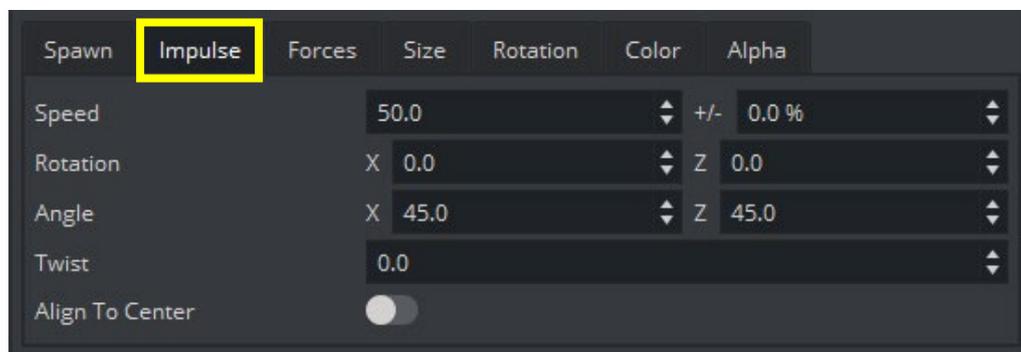
■ **Spawn** : パーティクルを放出する領域の形状とサイズを定義



Type	立方体 / 球 / コンテナ（平面） / 画像（グレースケール）から形状タイプを指定
Size	X / Y / Z 値から 選択したサイズを設定

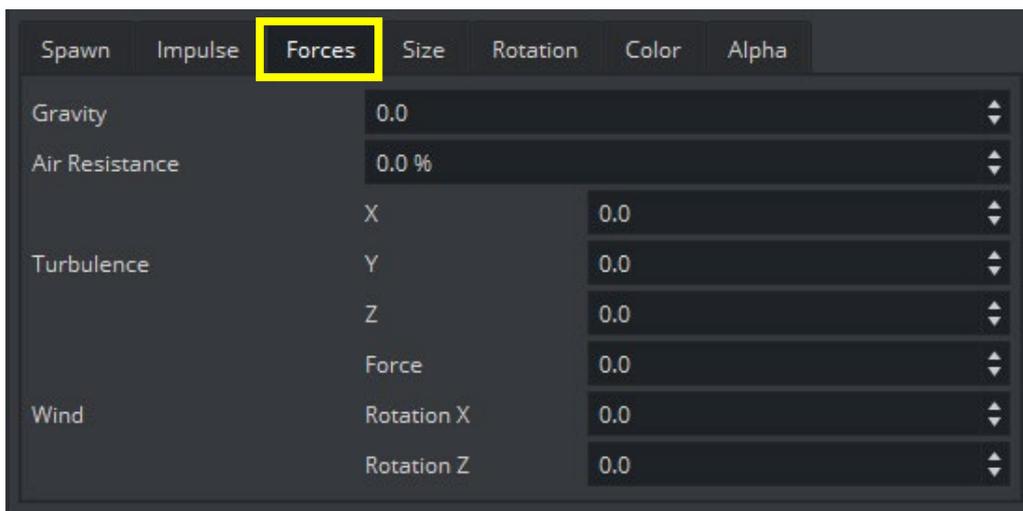
※選択した Type により変動

■ **Impulse** : パーティクルの放出方向や速度を定義



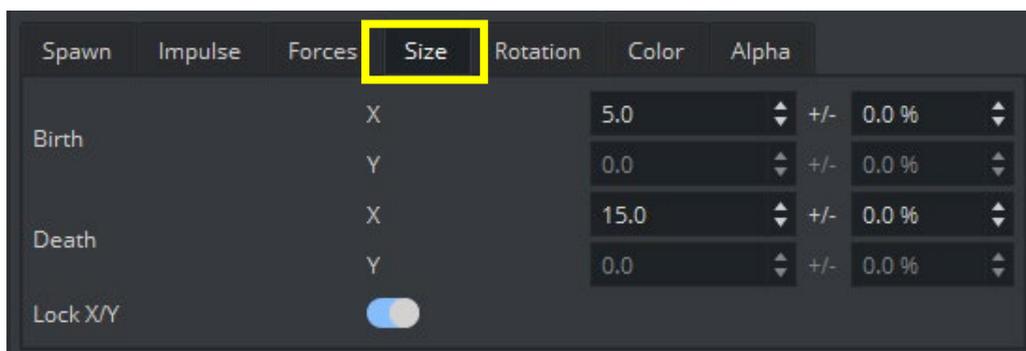
Speed	パーティクルが放出される時の速度を設定 +/-で速度のばらつきを設定（%）
Rotation	X / Z 軸から 放出する方向を設定
Angle	X / Z 軸から 放出する幅を設定
Twist	Y 軸から 放出する方向を設定
Align To Center	上記のすべてのパラメータを無効にし、パーティクル中央から放射状に放出されます

■ Force : 重力や風、空気抵抗など環境効果を定義



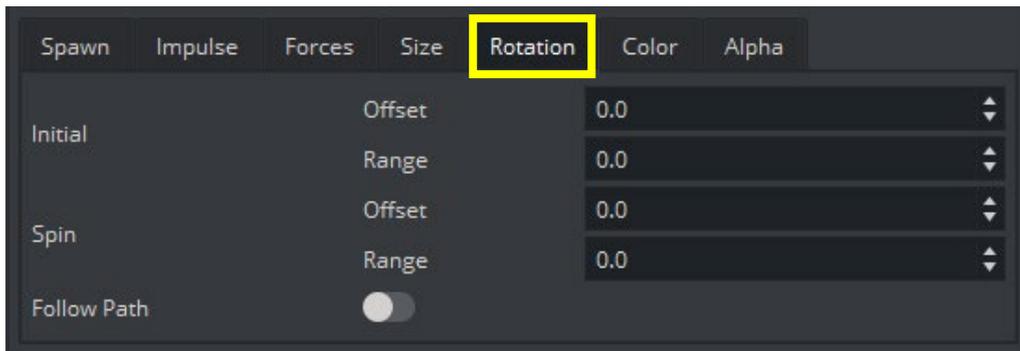
Gravity	重力の度合いを設定 数値が負の場合は粒子は上昇します
Air Resistance	空気抵抗の度合いを設定 初速から減速します
Turbulence	X / Y / Z 軸から乱流を設定 動きとしてはジッターリングに近い
Wind	Force で風の強さを設定 X / Z 軸から方向を設定

■ Size : パーティクルの誕生と消滅時の粒子サイズを定義



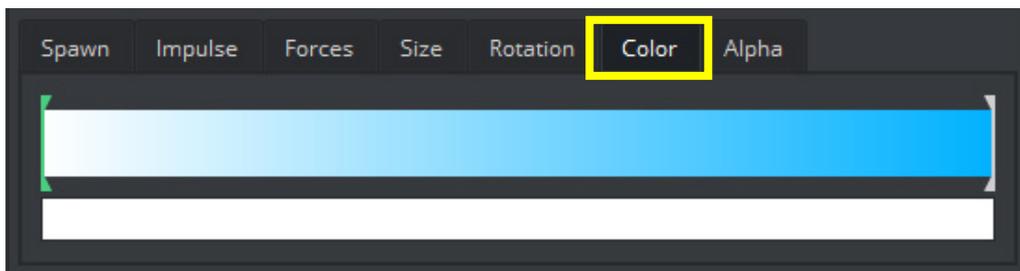
Birth	誕生タイミングでの X / Y サイズを設定
Death	消滅タイミングでの X / Y サイズを設定
Lock X/Y	軸をロックまたはロック解除

■ **Rotation** : パーティクルの角度及びスピナーアニメーションを定義

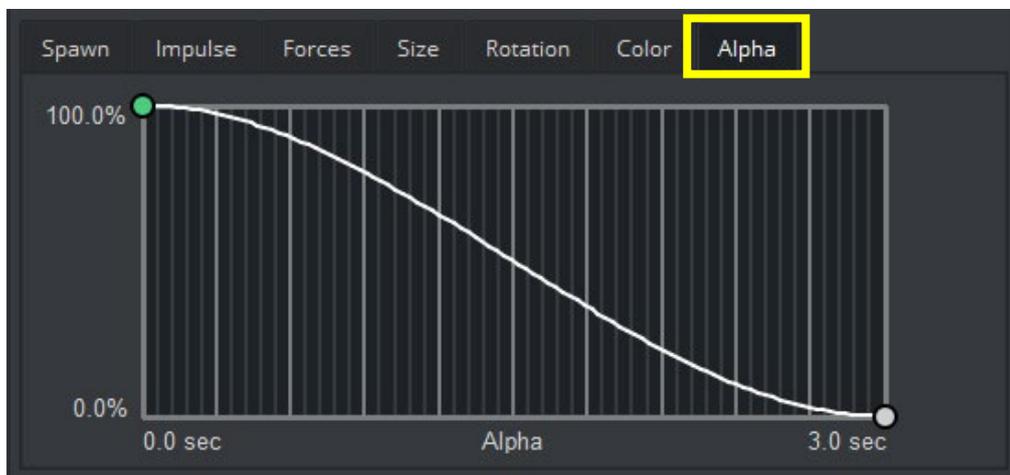


Initial	パーティクルの角度及び、角度範囲を設定
Spin	スピンの度合い及び、回転角度の範囲を設定
Follow Path	粒子の角度が、常に移動する方向に回転する

■ **Color** : 左側（誕生）から右側（消滅）への色変化を設定



■ **Rotation** : 左側（誕生）から右側（消滅）への透明度の変化を定義



2. パーティクル (RfxSmoke) モーションコントロール

RfxCollider Target / Source



両プラグインを組み合わせ使用し、パーティクルに当たり判定を付与します

RfxSmokeのコンテナへTarget、衝突するオブジェのコンテナへSourceを追加します

衝突オブジェクトの形状は、プリミティブの球体及びボックスのみ

RfxLatLong



“緯度”と“経度”を指定し、球体状にパーティクルを展開します

主に惑星等の球の表面にフィットさせるのに活用出来ます

RfxMagnet



磁場に似たモーションパターンを展開します

X / Y / Z軸で磁場の位置を指定し、パーティクルに対する引力（斥力）を設定します

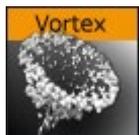
RfxTurbulance



乱流のような力をパーティクルに適用します

これを使用して風のような効果や、ランダムな方向の変化を作成できます

RfxVortex



パーティクルに渦のような効果を適用します

X / Y / Z軸で渦の中心を指定し、竜巻や台風のような効果を作成できます

3. 実習：パーティクル演出 “メタルチップ”

■ RFXSmoke における懸念事項

より自然且つ効果的なメタルチップ演出を制作する際に、RFXSmoke の機能的に以下問題点がある

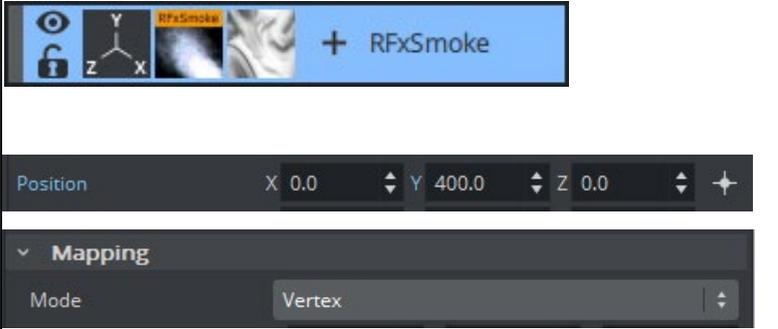
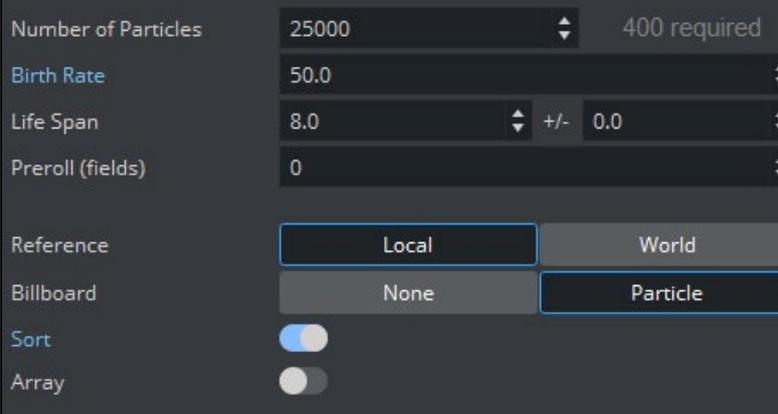
- ① 粒子の質感は 2D 画像のテクスチャーでみの設定される
- ② 粒子のスピンの設定は Z 軸の回転のみ

一方で、今回作成するメタルチップは…

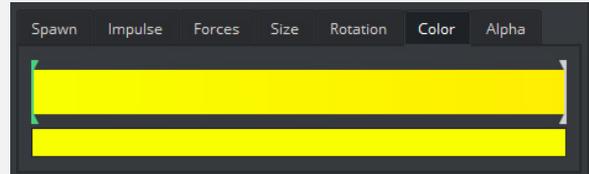
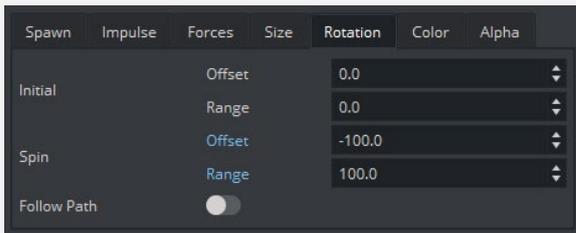
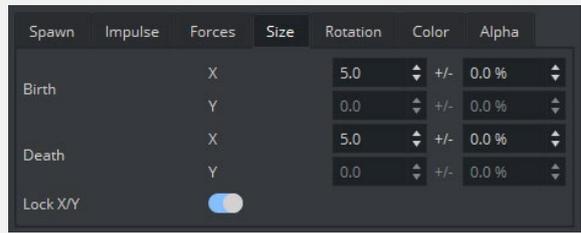
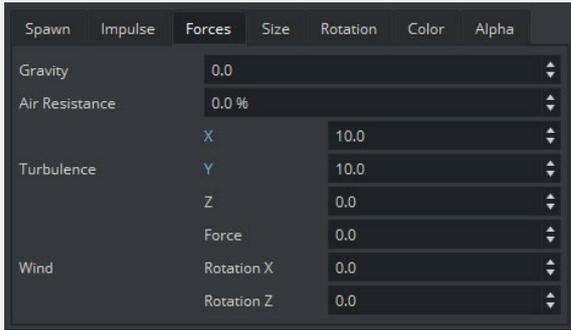
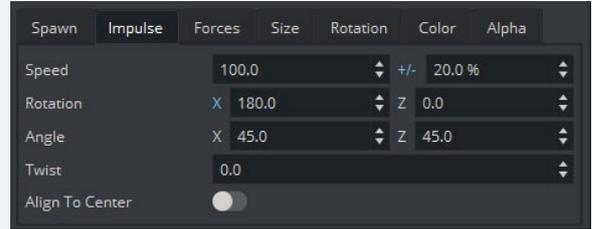
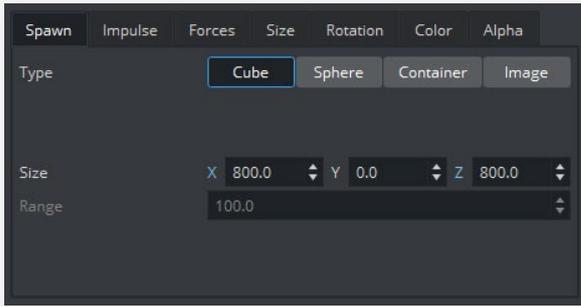
- ① 回転と共にキラキラと反射する質感
- ② X / Y / Z 軸、ランダムに回転しながら降る

以上のように、機能的に実現が難しい所をクリアーする必要がある

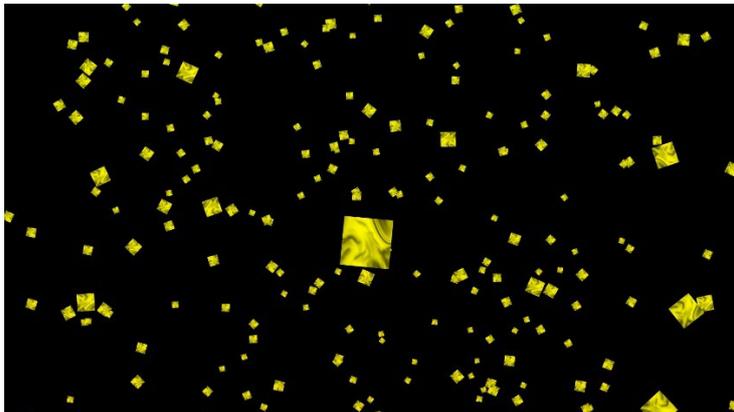
まず、そのまま”それっぽく”作成してみましょう

1		Tree へ RFXSmoke を入れ Position を上部へ移動 メタルっぽいテクスチャーを貼りつけ Mapping を Vertex へ変更
2		RFXSmoke のプロパティを 図のように変更

2



3



左記イメージとなり
反射する質感やランダムな回転は
実現できていないと思います

■ グラフィック(GFX)チャンネルを使用する



GFX チャンネルを使用すると、複数の独立したシーンのレンダリング結果を新しいシーンに対しテクスチャとして使用する事ができます。

この特性を利用して、動的なテクスチャを Vizrt 内で完結させ、①②の懸念事項を解消します

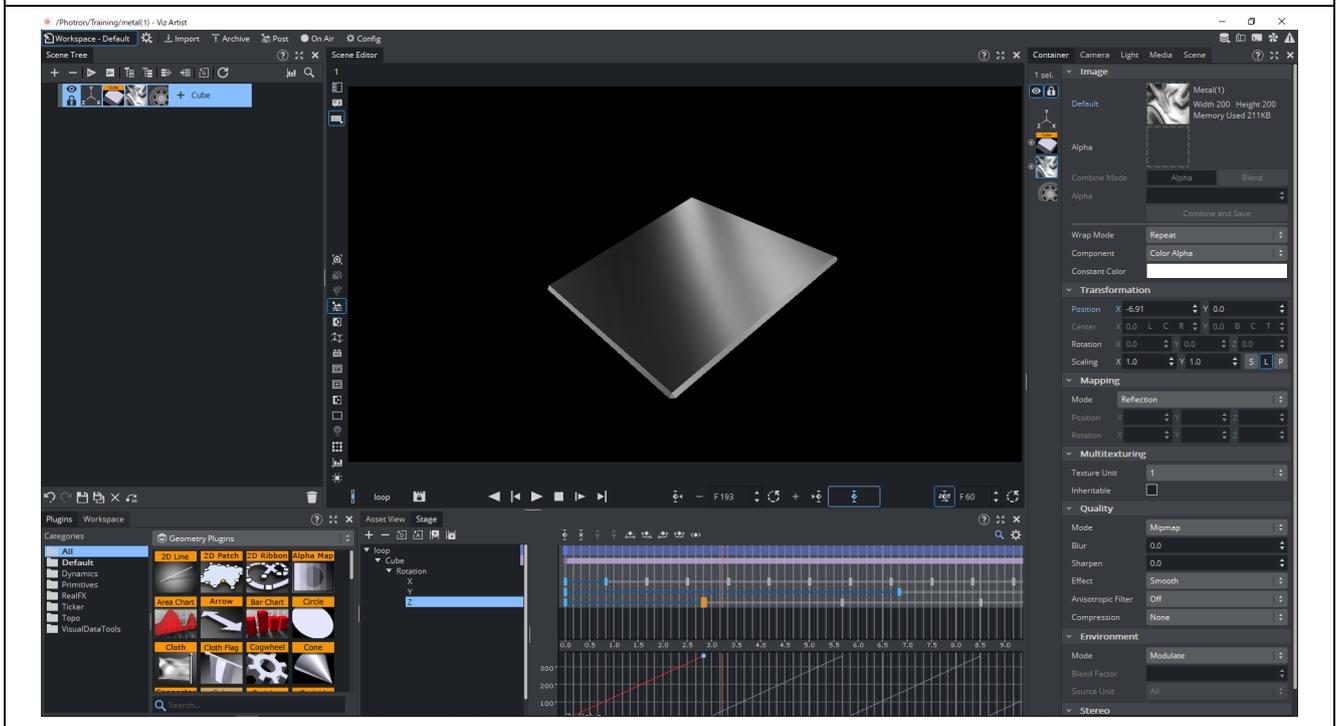
現在のパーティクルシーンを保存し、新規シーンでループ回転するメタルチップを作成します

注 1 : ディレクターを Auto Start 設定

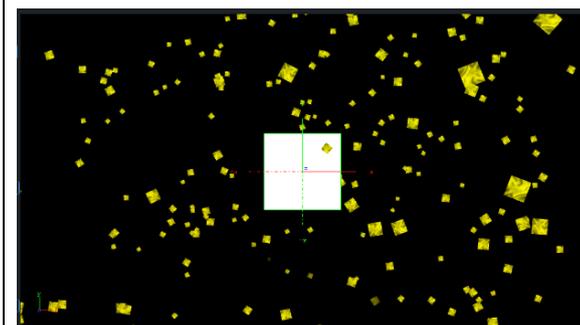
注 2 : Auto Key 若しくは Key プラグインにより、オブジェクトにキーが設定されているか確認

POINT

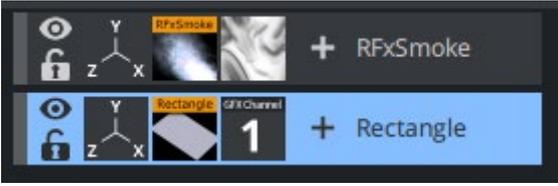
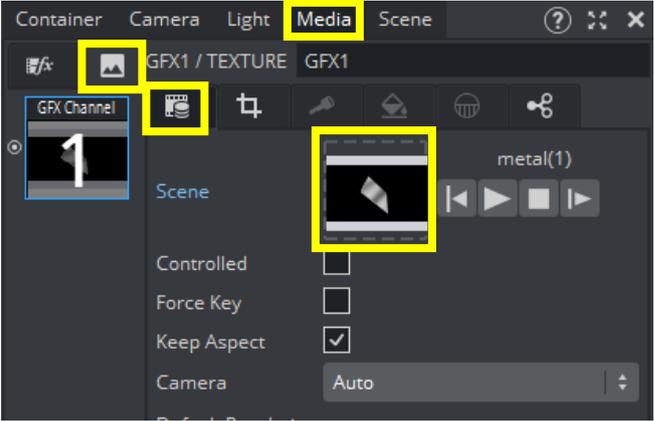
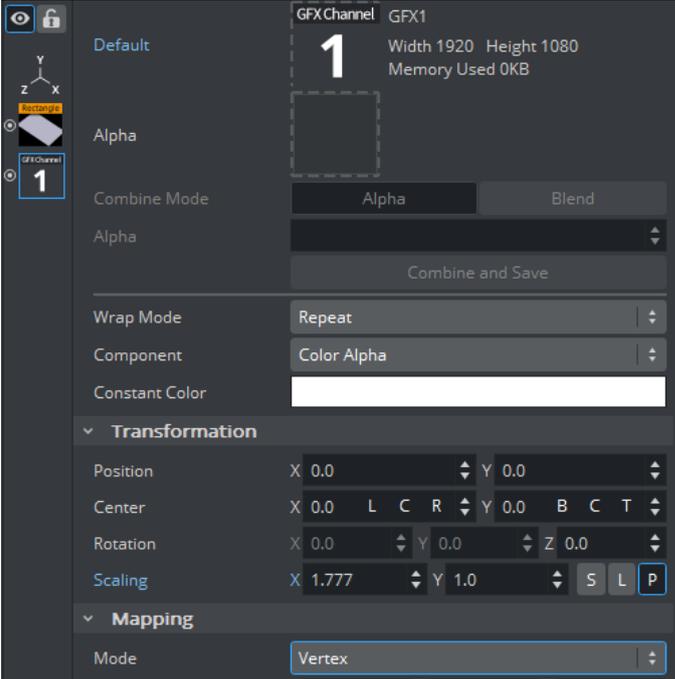
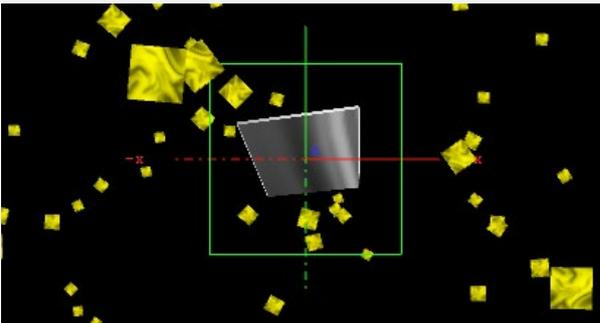
回転軸を Split Channel して X / Y / Z 毎に分解し、よりランダム感なアニメーションへ

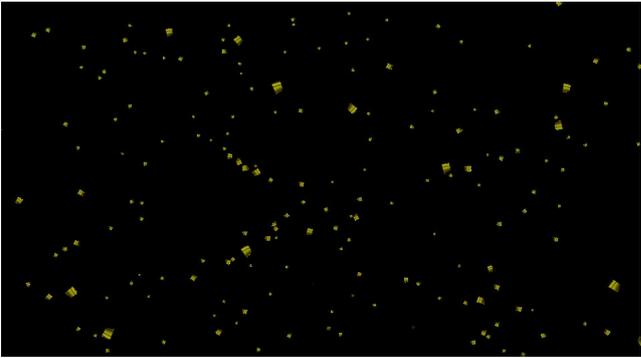


完成したらメタルチップのシーンを保存し、再度パーティクルのシーンを開きましょう



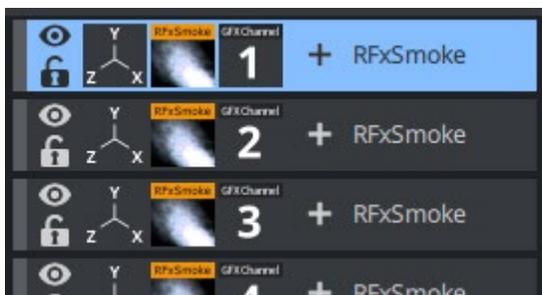
GFX チャンネルにて作成したメタルチップのシーンを呼び出しますが、調整しやすくするために Rectangle を Tree に追加しておきましょう

1		<p>Rectangle へ GFX Channel 1 を付与</p>
2		<p>Media タブの GFX Channel を選択し Scene のボックスへ作成したメタルチップのシーンをドラッグ & ドロップ</p> <p>メタルチップのシーンが読み込まれ、Rectangle へ動的テクスチャーが貼られます</p>
3		<p>GFX Channel のテクスチャーのプロパティを変更します</p> <p>パーティクルスプライトの比率は 1:1 なので、16:9 でレンダリングされたテクスチャーを調整するために、Scaling の X へ 1.777 を入力しています</p> <p>Mapping モードを Vertex へ変更</p>
4		<p>これでパーティクル用 GFX テクスチャーの準備が整いました</p>

5		<p>Rectangle 自体は非表示にしつつ GFX テクスチャーを RFXSmoke へコピー</p>
6		<p>質感が向上し、Z 軸以外の回転軸も付与されたように見えるかと思えます</p>

ここで更なる懸念事項として、回転及びリフレクションの反射タイミングがすべて同じタイミングになっている為、**全体的に違和感**が発生しているかと思えます。

これを解消するためには、GFX 側の回転アニメーションのパターンを増やす必要があります。



GFX チャンネルは 1 ~ 32 まで設定することが出来るので、アニメーション違いのシーンを複数作成して GFX を増やすパターンも考えられますが、複数シーンによる**メモリ増加**や**パフォーマンスへの影響**、制作作業を考えると**非効率**かと思えます

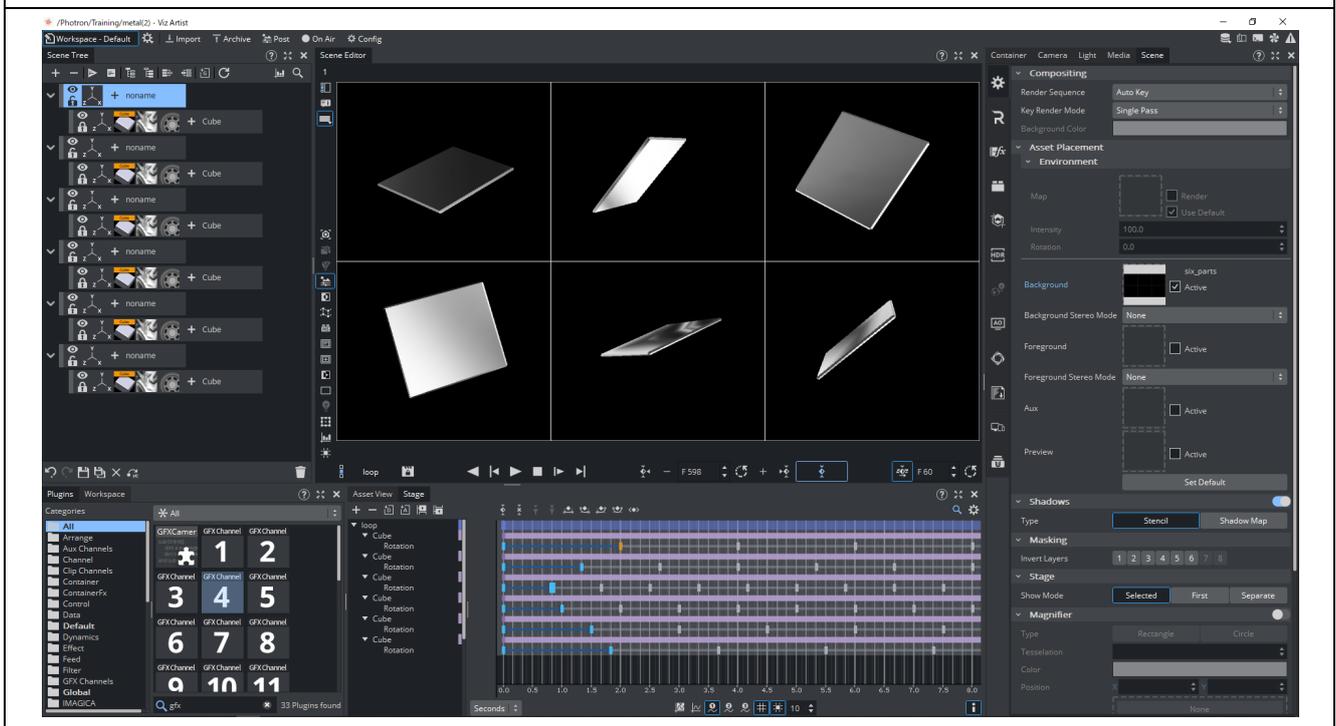
粒子毎にはある程度低解像度で問題ないので、アニメーション違いを一つのシーン(GFX)にまとめてみます

現在のパーティクルシーンを保存し、再度メタルチップのシーンを開きましょう
今回はメタルチップを6パターンのアニメーションを用意します

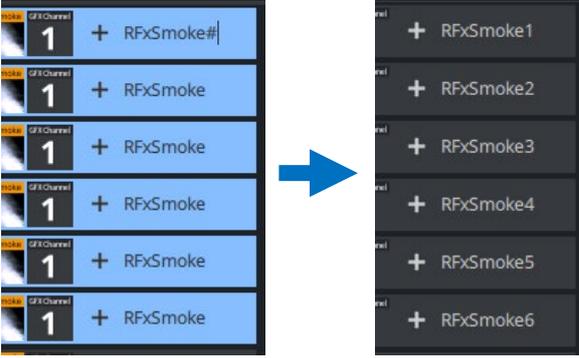
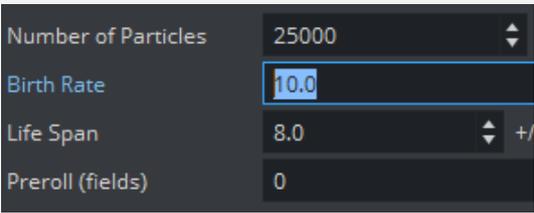
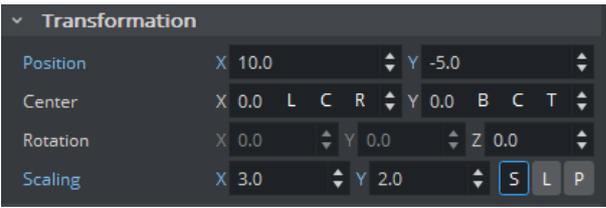
- ① : メタルチップのコンテナに親を作り、そのコンテナで位置を変更し6つを並べましょう
6分割したガイドテキスト等を用意しておくとう便利です
- ② : 各親コンテナで角度を変えつつ、アニメーション速度を調整し似たものが無いように調整

POINT

回転角度によって極端に細く見えたりして“悪目立ち”タイミングが発生するとしてしまうと複製感が強く出てしまうので、回転軸をX軸のみ等、**1軸だけの回転を推奨**します。



完成したら、**ガイドテキストを外し**メタルチップのシーンを保存し、再度パーティクルのシーンを開きましょう
別名保存した場合は、GFXチャンネルに改めて新しいシーンを設定し直してください

1		<p>6 分割の GFX テクスチャを Rectangle で確認しながら RfxSmoke へ割り振っていきましょう</p> <p>先に 6 パターン分 RFXSmoke を複製しておきましょう</p> <p>POINT 複数選択のリネーム時に“#”を入力すると自動的に昇順数字が入力され便利です</p>
2		<p>パーティクルの総数が単純に 6 倍になったのでコンテナを複数選択し、一括で Birth Rate を調整します</p>
3		<p>先ほどと同様、Rectangle で確認できた GFX テクスチャを RFXSmoke へコピーしていきます</p> <p>左図は左上部分のメタルチップ設定</p>

【参考】6 分割 テクスチャ-Position 設定値		
X:10 / Y:-5	X:0 Y:-5	X:-10 / Y:-5
X:10 / Y:5	X:0 / Y:5	X:-10 / Y:5
共通 Scaling X:3 / Y:2		

以上でランダムな“メタルチップっぽい”パーティクルが表現できたかと思えます

グループ化して複製し、Color を変更して更にパターンを増やしてみましょう！

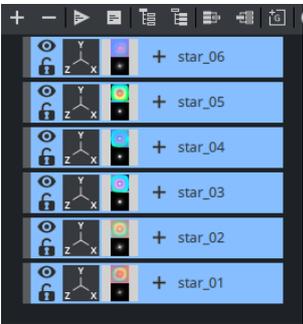
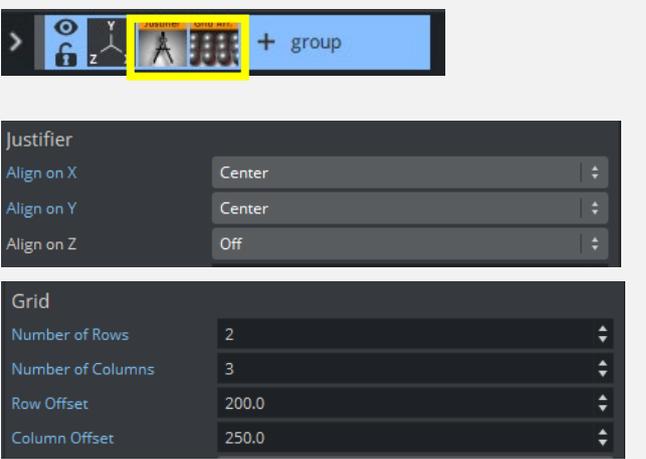
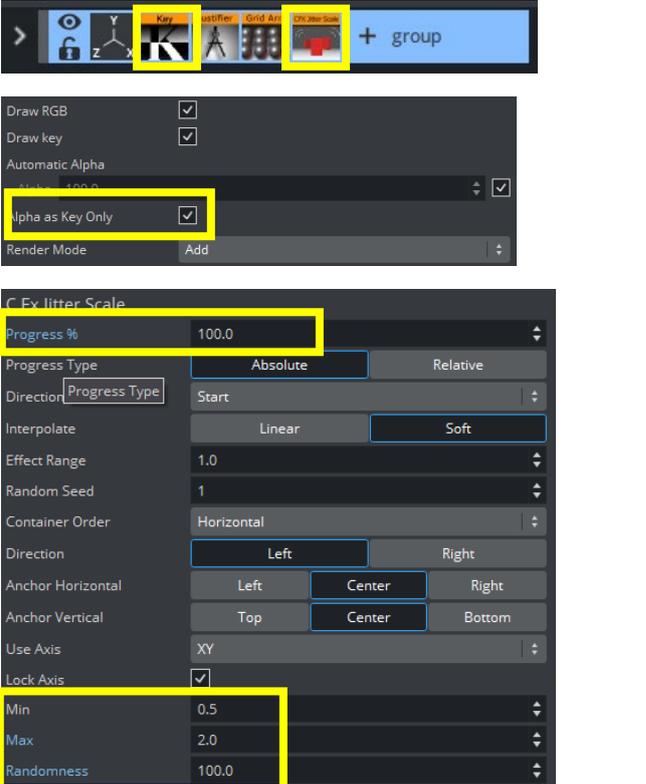


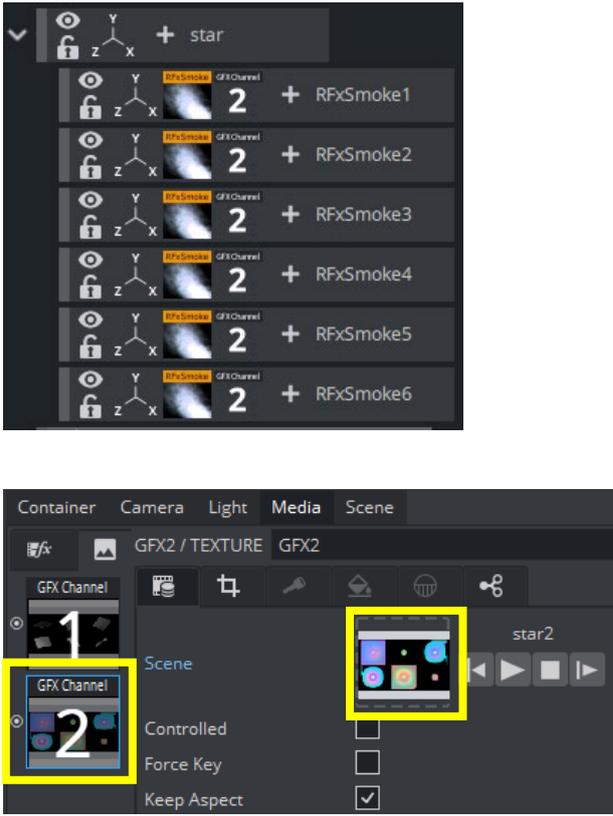
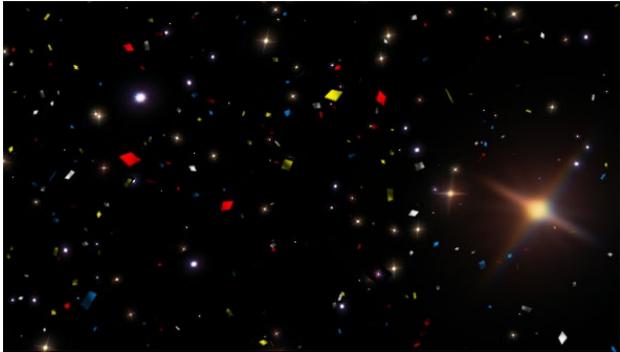
更にパーティクルの総量が増えるので、Birth Rate を調整しましょう

■ 別のグラフィック(GFX)チャンネルを追加

新しい6分割のシーンを作成して、2つ目の GFX チャンネルを追加してみましょう

簡易的に光がランダムに点滅しているパーティクルを作成してみます

1		<p>Classic モードで新規シーンを作成します</p> <p>Asset から6種類の"光"テクスチャーを Tree へ入れます</p>
2		<p>グループ化し、親に"Justifier"と"Grid"プラグインを付与</p> <p>それぞれ左記のように設定すると6つのテクスチャーが整列できます</p>
3		<p>続いて"Key"と"CFX Jitter Scale"プラグインを付与</p> <p>それぞれ左記のように設定します</p> <p>Jitter プラグインの影響で光がランダムに点滅しているようになったかと思えます</p> <p>Alpha as Key Only のチェックを入れたのは GFX チャンネルで他のシーン内で合成する際に半透明部分等が背景色を拾ってしまうので RGB を 100% で表示させます</p> <p>完成したら保存しましょう</p>

<p>4</p>	 <p>The top screenshot shows a list of six RFXSmoke objects, each with a 'GFX Channel' of 2. The bottom screenshot shows the 'Media' tab with a scene named 'star2' selected, and a 'GFX Channel' of 2 highlighted in a yellow box.</p>	<p>前項同様、複製してから GFX Channel 1 を GFXChannel 2 へ差し替えます</p> <p>Media タブの GFX Channel 2 を選択し Scene のボックスへ新規に作成した点滅シーンをドラッグ & ドロップ</p>
<p>5</p>	 <p>The screenshot shows a dark space scene with a bright star and numerous colorful particles (red, blue, yellow, white) scattered around it.</p>	<p>以下好みですが、光点滅用に値を調整</p> <ul style="list-style-type: none"> Life Span = 16 Impulse / Speed = 50 Force / Turbulence X & Y = 0 Size / Birth & Death = 15 +/-30% Color = White

以上のように、GFX Channelを駆使しつつ、RFXTurbulanceやRFXMagnet等のモーションコントロールエフェクトと組み合わせる事で、Vizrtでのパーティクル演出の幅が広がると思います

4. 実習 : Classic と Viz Engine Rendererの共存と制御

Vizrt 最新バージョンには2つのレンダリングモードが存在し、各シーン毎に選択する必要があります
双方でシェーディングが異なるために、マテリアル等の違いやプラグインに相違があり、例えば…

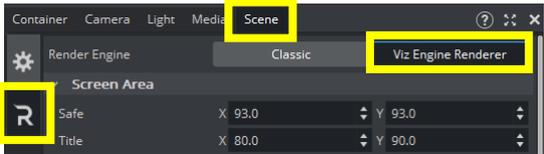
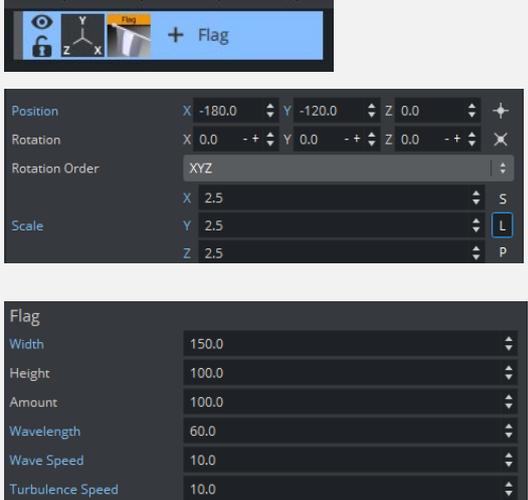
- Classic 用のマテリアル、VER 用のマテリアルはお互いに機能しない
- テクスチャーの貼り方が異なる
- パーティクルは Viz Engine Renderer では使用できない
- VER は PBR マテリアルやポストエフェクトの機能により、リッチなシーン制作が可能 etc.

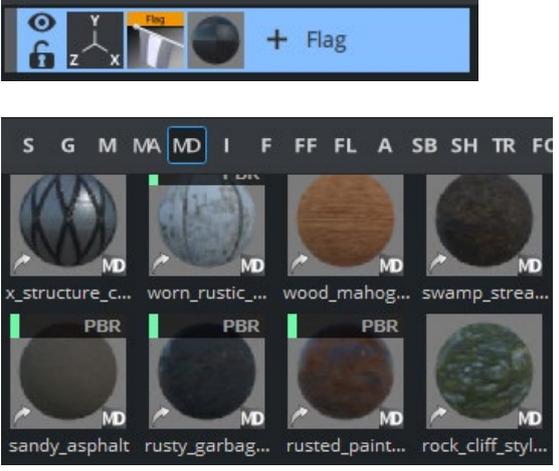
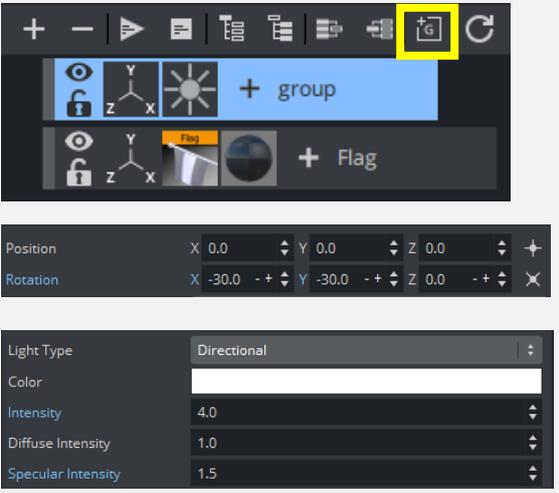
状況により一長一短という事もあり、コンテンツ内容によっては両モードを共存させなければならない時も…
そのような時にはグラフィック（GFX）チャンネルを使用することにより、レンダリングモードに左右されず
複数のシーンを合成する事が出来ます

制御に関しまして、今回は **Control Object** での制御を少しですがご案内いたします

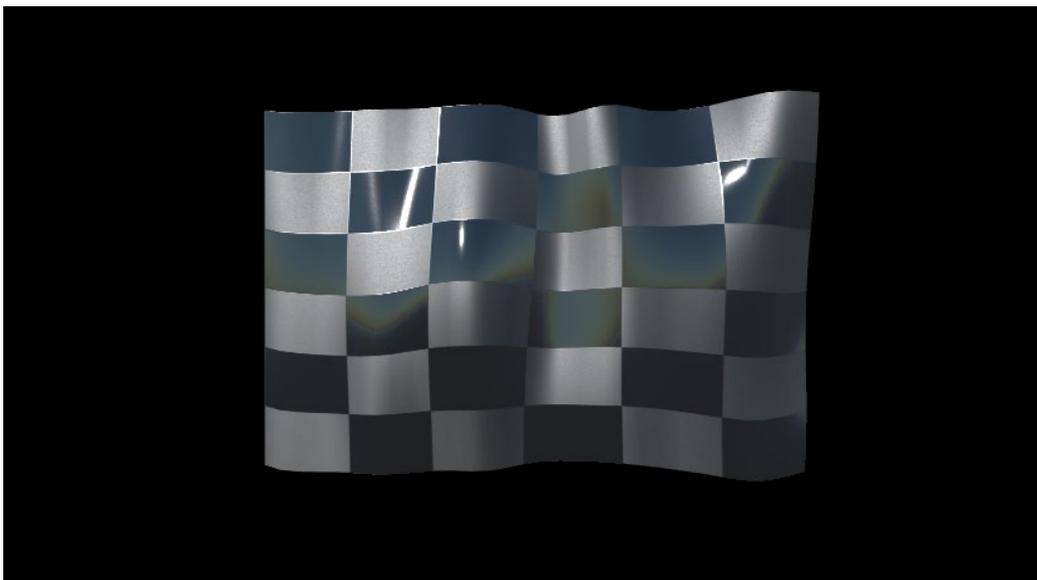
■ Viz Engine Renderer でのシーン制作

簡単ではありますが、PBR マテリアルで質感付けされた “旗” を作成します

1		Viz Engine Renderer モードで新規シーンを作成 します まず Render Engine が正しい事を確認
2		Flag プラグインを Tree へ入れます 位置や大きさ、Flag のプロパティを調整しましょう 左図設定値は参考まで…

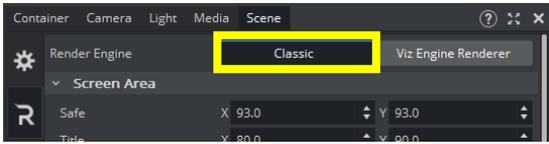
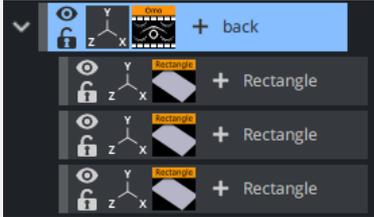
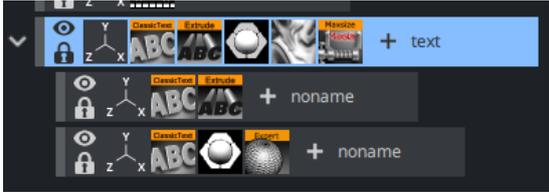
<p>3</p>		<p>質感を付与しましょう</p> <p>新規で PBR マテリアルを作成するでも良いのですが、今回はいくつかご用意したサンプルから付与します</p> <p>PBR マテリアルは Aset View の"MD"でソート</p>
<p>4</p>		<p>合わせてライトも配置しましょう</p> <p>VER モードではシーンベースではなく、コンテナベースでライトを配置・設定します</p> <p>空のグループコンテナを作成し Light プラグインを付与 左図設定値は参考まで…</p>

例)



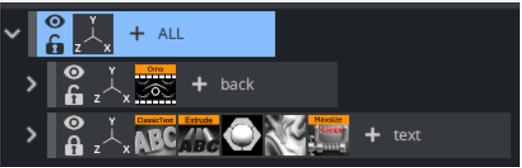
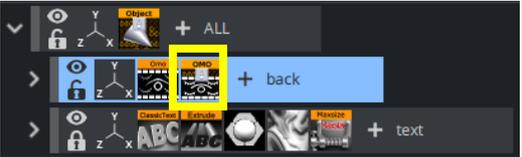
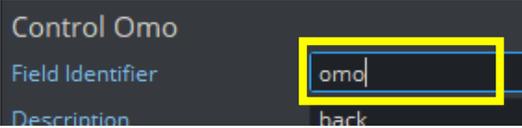
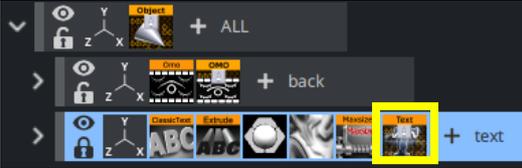
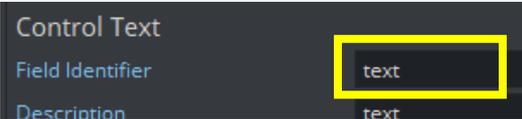
■ Control Object 制御 Classic シーン制作

“テキスト” と “OMO”を Control Object で制御できるシーンを作成します

1		Classic モードで新規シーンを作成します
2		<p>背景を作成します</p> <p>Rectangle を Tree へ入れ、奥へ移動させてから Screen Size 右側のボタンを押し、スクリーンにフィットさせます</p>
3		<p>Rectangle に色を加えます</p> <p>左図は Rectangle のプロパティで色付けした値です マテリアル / テクスチャでも構いません</p>
4		複製しながら“色違い”を3種類作成しグループ化して“Omo”プラグインを付与
5		<p>続いてテキストオブジェクトを作成します</p> <p>フォントと Extrude プラグイン、テクスチャ等を組み合わせて 3D テキスト作成してみましょう</p> <p>MaxSize プラグインを使用して、レンダリング領域から出ないように設定してください</p>

例)



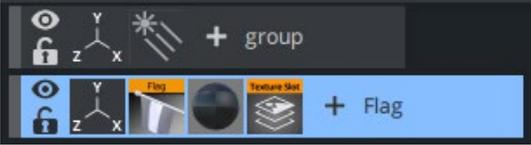
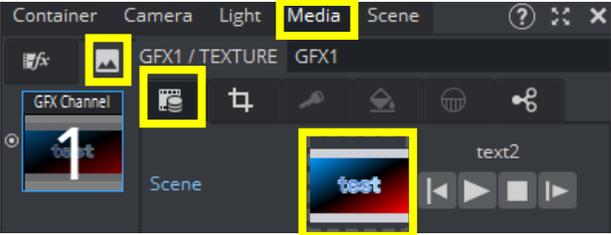
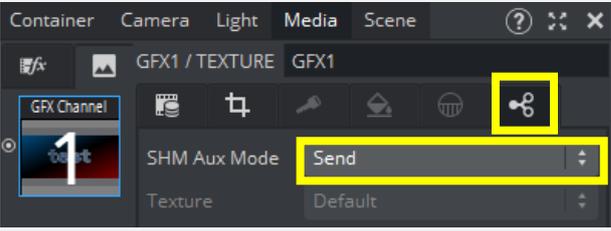
6		<p>Control Object の設定をします</p> <p>すべてのコンテナの親を作成し任意の名前を設定 今回は "ALL" としておきましょう</p>
7		<p>Omo プラグインを付与したコンテナへ ControlOmo プラグインを追加します</p> <p>自動的に最上コンテナへ ControlObject プラグイン が追加されます</p>
8		<p>ControlOmo のプロパティ Field Identifier へ任意の文言を指定しましょう</p> <p>これが制御する際の識別"ID"となります 今回は"omo"としておきましょう</p>
9	 	<p>続いてテキストのコンテナ（親のみ）へ ControlText プラグインを付与</p> <p>ID を"text"としておきましょう</p>
10		<p>上記設定された ControlText プラグインを 子のテキストコンテナへコピーして付与します</p>

設定が完了したら Control Object ウィンドウから動作の確認をします
Control プラグインの Fields 毎で項目に表示されるので、それぞれ Value の値を変更してみましょう
Fields"text"にある "i" マークは同名の ID が複数ある場合に表示されるマークです

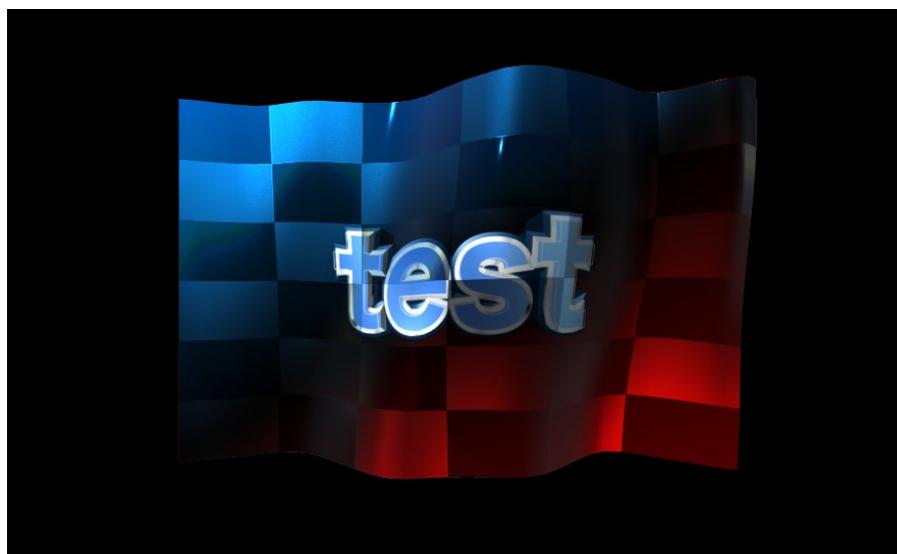
Fields	Description	Value	Type	Minimum	Maximum
omo	back	0	Integer	0	1000
i text	text	test	Formatted Text		

■ GFX チャンネル設定

VER モードの“旗”のシーンへ Classic シーンをテクスチャーとしてロードします

1		<p>旗のシーンを開きます</p> <p>PBR マテリアルにテクスチャーを追加するために TextureSlot プラグインを付与</p>
2		<p>TextureSlot プラグインのプロパティを開き Texture のボックスへ GFX Channel 1 をドラッグ&ドロップ</p>
3		<p>Media タブの GFX Channel を選択し Scene のボックスへ作成したテキストのシーンをドラッグ&ドロップ</p>
4		<p>続いて Aux のタブ内 SHM Aux Mode を “Send”へ変更します</p>

例)



■ GFX チャンネルの制御

GFX チャンネルへロードされたシーンの Control Object を制御します

まず ControlObject を制御するための基本コマンドは…

RENDERER*TREE*\$①*FUNCTION*ControlObject*in SET ON ② SET ③

①…ControlObject プラグイン付与コンテナ名 ②…ID ③…Value

TEXT でも OMO でも同じコマンドとなり、シーン構築やコントロールパネル制作等の管理がしやすくなります
先ほどのシーンと照らし合わせると…

RENDERER*TREE*\$ALL*FUNCTION*ControlObject*in SET ON omo SET ③

RENDERER*TREE*\$ALL*FUNCTION*ControlObject*in SET ON text SET ③

上記となり、③の Value へ任意の値を設定します

GFX Channel へ指定したシーンへの制御コマンドに関して"REND"や"THIS_SCENE"の後
*GFX*1*でチャンネル番号を指定する事で各種セットやディテクタースタート等、通常の制御が可能です

RENDERER*GFX*1*STAGE*DIRECTOR*XXX START

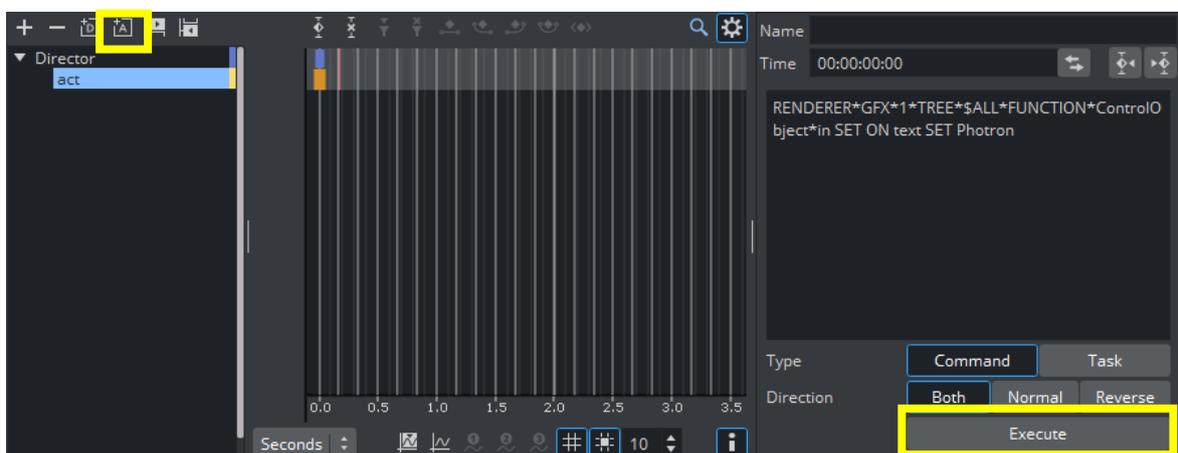
THIS_SCENE*GFX*1*TREE*@XXX*GEOM*TEXT SET XXX

以上を加味し、最終的な制御のコマンドは…

REND*GFX*1*TREE*\$ALL*FUNCTION*ControlObject*in SET ON omo SET ③

REND*GFX*1*TREE*\$ALL*FUNCTION*ControlObject*in SET ON text SET ③

最後に Stage に Action キーを作成してコマンドを実行してみましょう



【 MEMO 】