

## HOUDINI FOUNDATIONS

# 破壊 FX

ビジュアルエフェクトが楽しい理由の1つは、現実には何も破壊せずに、さまざまなものを爆破できることです。このレッスンでは、パーティクルの火花を使用して導火線に火を付け、カートゥーンの爆弾を爆発させます。爆弾にはリジッドボディダイナミクス、火と煙には Pyro FX を使用します。このレッスンでは、さまざまなシェルフツールやネットワークノードを使用して、ダイナミックシミュレーションをセットアップする方法を学びます。

作成するショットを完全に理解するために、すべての要素をゼロから作成して、エフェクトをシミュレートします。Houdini シーンのままさまざまなコンテキストで、シミュレーションノードがどう機能するかを理解しやすくなるはずです。最後に、Karma レンダラを使用してショットをレンダリングします。



### ACES | OPENCOLORIO セットアップ

Pyro FX の使用時、より正確にカラーを表示するには、**Academy Color Encoding System (ACES)** を使用します。そのためには、Scene View の **ビューポート (Persp)** メニューから、**Correction ツールバー** を表示します。右にある矢印ボタンから、**OpenColorIO** を選択します。これにより、ディスプレイが **sRGB**、出力が **SDR Video - ACES 1.0** になります。この設定は現在のセッションにのみ有効で、Houdini を開くたびに戻す必要があります。



### レッスンの目標

爆弾をモデリングし、パーティクルの火花、リジッドボディダイナミクス、Pyro FX を使用して爆発させます。

### 学習内容

- 爆弾をモデリングし、導火線をアニメートする方法
- カメラをアニメートし、ショットをセットアップする方法
- すずの軌道(トレイル)と導火線の火花をセットアップする方法
- 爆弾のジオメトリを粉砕して爆発させる方法
- 爆発用の Pyro FX シミュレーションをセットアップする方法
- マテリアルとテクスチャをセットアップする方法
- Solaris コンテキストで Karma を使用してエフェクトをレンダリングする方法

### 使用する機能とソフトウェア

Houdini 19.5+ の機能を前提として、書かれています。

このレッスンの手順は、以下の Houdini 製品で実行可能です。

Houdini Core	×
Houdini FX	✓
Houdini Indie	✓
Houdini Apprentice	✓
Houdini Education	✓

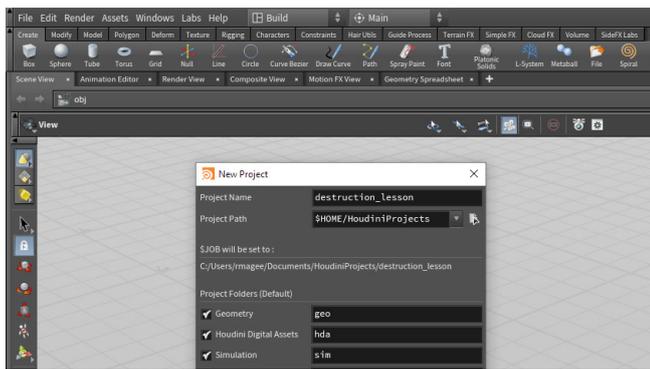
ドキュメントバージョン 1.0 | 2022 年 7 月  
© SideFX Software



## パート 1

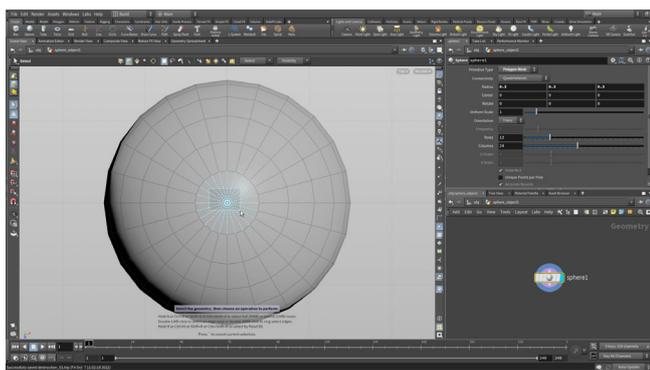
# 爆弾のモデリング

爆弾のジオメトリを作成するには、プリミティブの球から始め、上部を開口部に変更します。これにはポリゴンの押し出しとベベルを使用して、最終形状に必要なジオメトリを定義します。レッスンの後半で、この爆弾を砕きます。



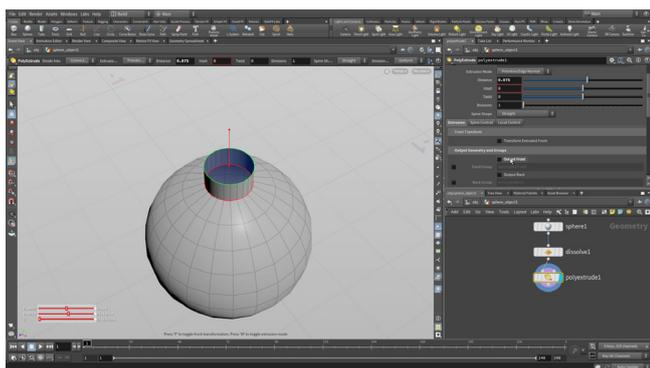
**01** **File > New Project** を選択します。Project Name を **destruction\_lesson** に変更し、**Accept** を押します。これにより、プロジェクトディレクトリとサブディレクトリが作成され、このショットに関連するすべてのファイルが格納されるようになります。

**File > Save As...** を選択すると、新しい **destruction\_lesson** ディレクトリが表示されます。ファイル名を **destruction\_01.hip** に設定し、**Accept** をクリックして保存します。



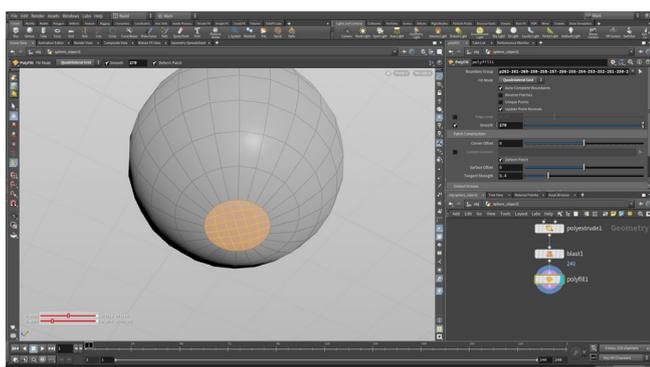
**02** ビューポートで、**C** を押して Radial メニューを表示します。このメニューから **Create > Geometry > Sphere** を選択します。ビューポートで、**Enter** を押して球を原点に配置します。上部のオペレーションコントロールツールバーで、**Radius** を **0.3, 0.3, 0.3** に設定します。

**S** を押して Select ツールにしたら、**3** を押してエッジ選択を呼び出します。**スペースバー + 2** を押して Top ビューに移動します。球の上部と下部のエッジを矩形選択し、**Delete** を押します。これによりエッジが削除され、2つの円形ポリゴンができます。**スペースバー + 1** を押してパースビューに戻ります。



**03** **S** を押して Select ツールにし、**4** を押してプリミティブ(フェース)選択にします。球の上部の円形ポリゴンを選択します。

**C** を押して Radial メニューを表示し、**Model > Polygons > PolyExtrude** を選択します。**Distance** を約 **0.075** にして、ハンドルを上に移動します。**Output Front** を **オフ** にします。



**04** 球の下部の円形ポリゴンを選択し、**Delete** を押します。これにより、ネットワークに **blast** ノードが追加されます。**S** を押して **Select** ツールに移動し、**3** でエッジ選択に切り替えます。先ほど作成した穴のエッジを **ダブルクリック** して、すべてのエッジを選択します。

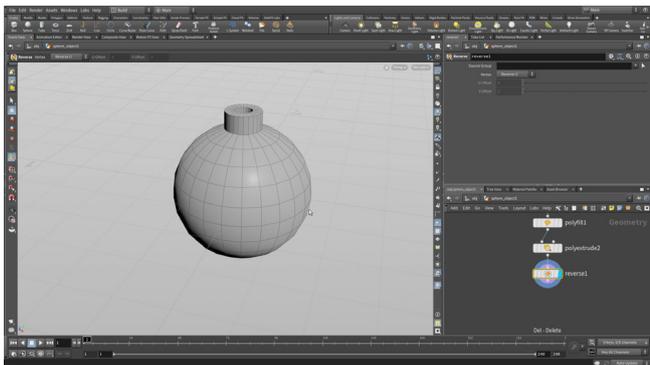
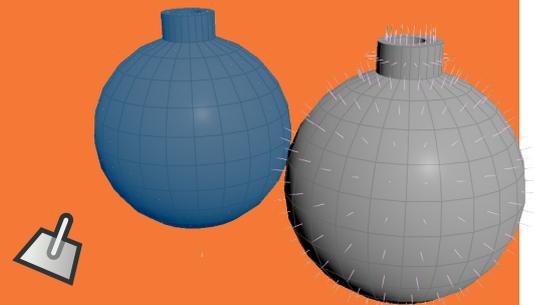
**Tab > Polyfill** を押します。パラメータエディタで、**Fill Mode** を **Quadrilateral Grid**、**Smooth** を **270** に設定します。球の下部に、単一のポイントに集中しない、クリーンなトポロジが作成されます。



## サーフェス法線

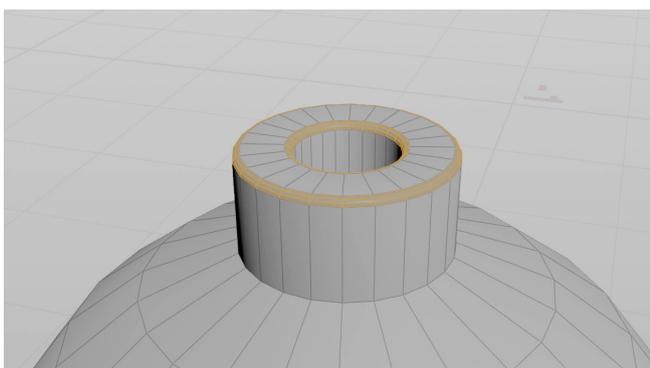
すべてのプリミティブに法線方向があり、片方の側が内側、もう片方の側が外側になっています。爆弾のジオメトリを押し出すと、最初はジオメトリが裏返しになります。これは、青色のフェースで示されます。その後、Reverse ノードを使用して法線の変更を行います。

サーフェス上に法線を表示するには、Scene View ペインの右側の Display Options バーにある Primitive Normals ボタンを使用します。



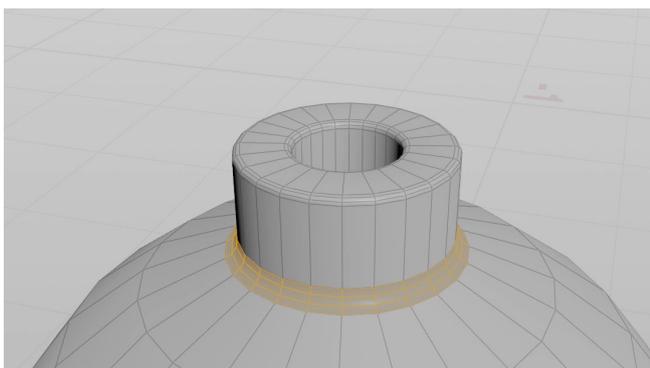
**05** N を押してすべてのフェースを選択し、再度 Polyextrude ツールを選択します。約 **-0.04** の Distance 値に押し出します。パラメータエディタの Extrusion タブで、Output Back をオンにします。

N を押してすべてのフェースを選択し、再度 Tab を押して Reverse と入力していきます。このノードは、すべてのポリゴンの法線を反転させます。これで、法線が正しい方向を向くようになりました。



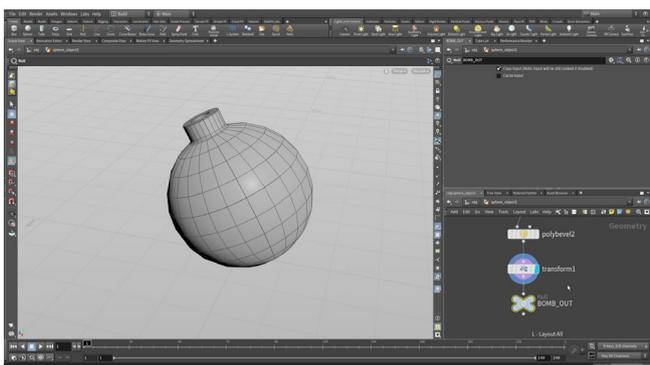
**06** S を押して Select ツールに移動したら、3 でエッジ選択に変更します。爆弾の上部のエッジをダブルクリックしたら、Shift を押しながらダブルクリックして上部の内側の円を選択します。

C を押し、Radial メニューから Model > Polygons > Polybevel を選択します。Distance を 0.005 に設定します。Shape を Round、Divisions を 3 に設定します。



**07** S を押して Select ツールに切り替えます。爆弾の円の部分と上部の押し出された部分が変わる箇所のエッジをダブルクリックします。

Q を押して最後に使用した Polybevel ツールに戻り、Distance を 0.01、Shape を Round、Divisions を 3 に設定します。



**08** ネットワークビューで、Tab > Transform を押して、それをネットワークの終端に追加します。Translate Y を 0.3、Rotate X を 27 度に設定します。

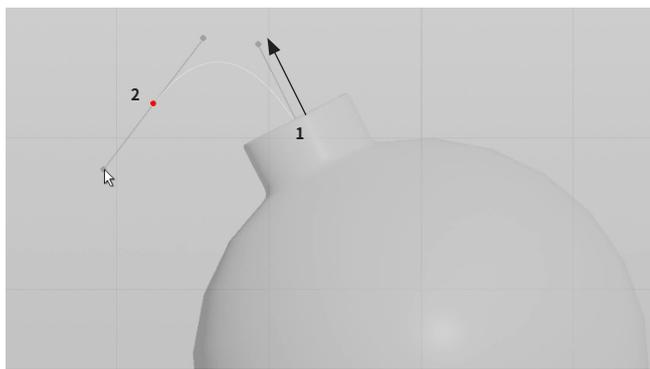
Null ノードを追加します。終端に null を接続し、null ノードに Display フラグを設定して表示します。null ノードの名前をダブルクリックして、BOMB\_OUT に変更します。

オブジェクトレベルに移動して、オブジェクトの名前を bomb\_geo に変更します (爆弾のジオメトリが含まれるため)。

## パート2

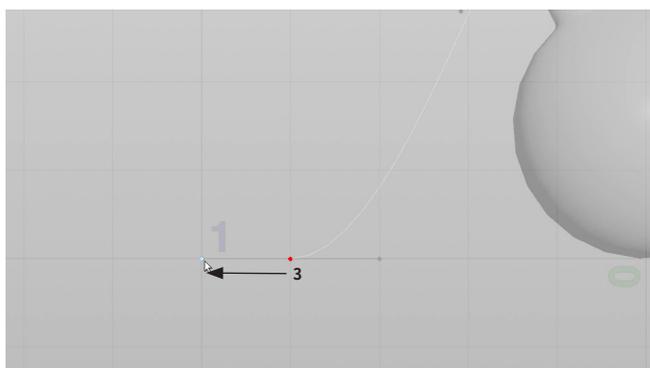
# 導火線のモデリング

導火線の作成は、爆弾の上部から伸びる Bezier カーブから始めます。地面にカーブをくねらせて、長い導火線を作成します。カーブの方向を逆にして導火線のアニメーションの準備をしてから、Polywire ノードを追加して導火線に厚みを与えます。

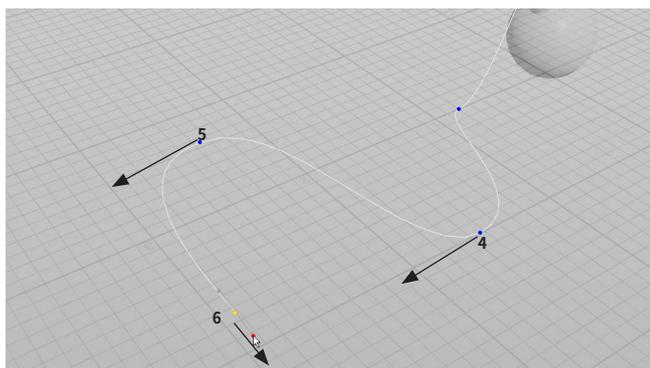


**01** **スペースバー + B** を押してすべての角度から爆弾を表示します。Right ビューに移動し、**スペースバー + B** を押して拡大します。

**C** を押して Radial メニューを表示し、**Create > Geometry > Curve** を選択します。**クリックアンドドラッグ**して、カーブの1つ目のポイントと接線ハンドルを作成します。次に、ポイントを追加し、**下にドラッグ**して下向きのカーブを簡単に描画します。



**02** **X** を押して **Grid** を選択し、グリッドスナップングをオンにします。地面で**クリックアンドドラッグ**して3つ目のポイントを作成し、接線ハンドルは地面に揃えます。



**03** **スペースバー + B** を押して4面ビューに戻ります。マウスをパースビューの上に移動し、再度**スペースバー + B** を押してビューを拡大します。

**グリッドスナップング**をオフにし、**Display Options** バーの上から2つ目のボタンを使用して、**コンストラクション平面**をオンにします。これにより、カーブが地面から離れなくなります。

接線をドラッグして新しい3つのポイントを描画し、カーブの形状を定義します。

**MMB クリック**してカーブを完成させます。



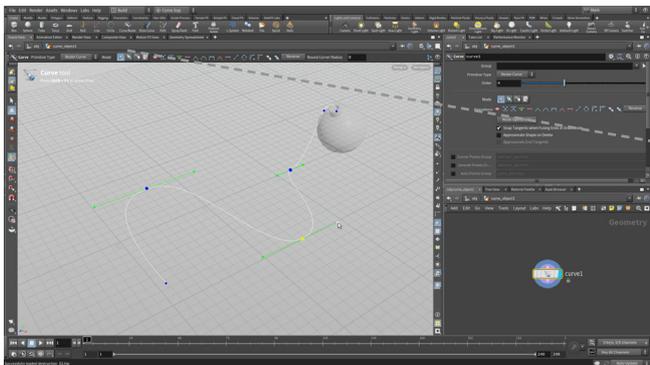
## ツールのヒント

Curve ツールにはツールのヒントが付属しており、作業時に Scene View に表示されます。このツールのさまざまなショートキーオプションが表示され、ツールの機能を理解するのに役立ちます。

**Shift + F1** を使用して折り畳むと、ツール名のみが表示されます。ヒント付きのツールは増えています。Houdini の今後のバージョンでは、さらに多くのツールでヒントが表示されるでしょう。

**Curve tool**  
Press **Shift + F1** to hide hints

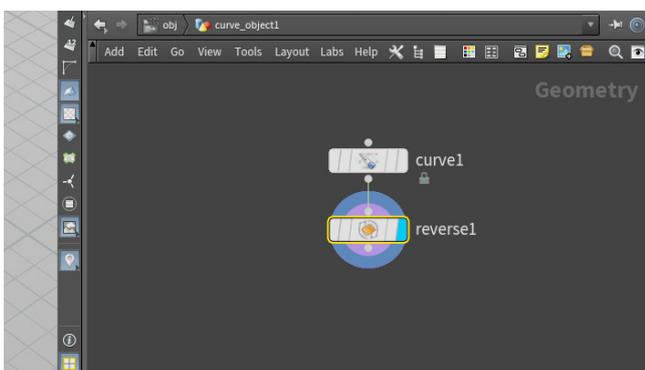
	Mode	Edit
<b>F / G / H</b>	Curve Type	<b>Bézier</b>
<b>Shift +</b>	Drag entire curve	
<b>Ctrl +</b>	Insert a point (on curve)	
<b>Ctrl +</b>	Change radius relatively (radius widgets)	
	Tangents	
<b>1 - 3</b>	Set selected point type	



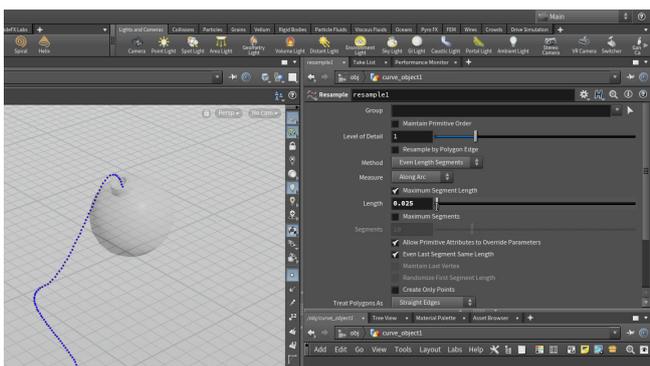
**04** オペレーションコントロールツールバーで、カーブの **Mode** を **Edit** に変更します。カーブ上の編集ポイントをクリックして、カーブの形状を微調整できるようになります。



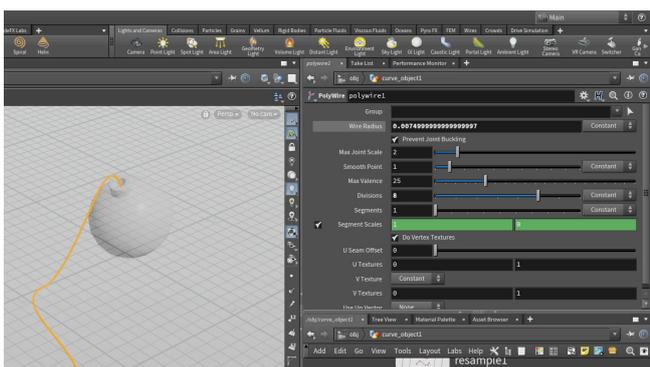
地面でポイントを選択して編集し、カーブを希望通りのルックにします。タンブルして、カーブが地面上にあることを確認します。



**05** Scene View で、**Tab > Reverse** を押します。**N** を押してカーブ全体を選択し、**Enter** を押します。カーブを爆弾から出てきたように描いたため、このままでは逆の方向からアニメートされてしまいます。これで、導火線の火が付く側から、カーブが開始できるようになります。



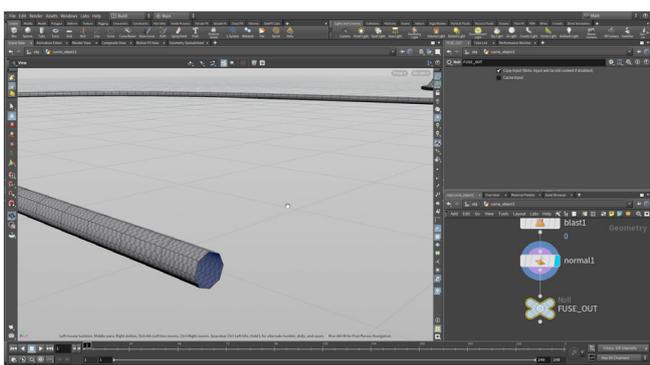
**06** **Display Options** バーの **Points** オプションをオンにします。**Resample** ノードを追加します。**resample** ノードで、**Maximum Segment Length** を **0.025** に設定して、ディテールを追加します。**resample** ノードによって、ポイントが均一に配置されます。



**07** **Polywire** ノードを追加して、ワイヤーに厚みを与えます。**Wire Radius** を **0.0075**、**Divisions** を **8** に設定します。

**reverse** ノードと **polywire** ノードの間に、**Transform** ノードを追加します。polywire ノードに移動し、**Wire Radius** パラメータを **RMB** クリックして、**Copy Parameter** を選択します。**transform** ノードに戻り、**Translate Y** を **RMB** クリックして、**Paste Relative References** を選択します。

これで、導火線全体が持ち上がり、地面のグリッドに半分埋まっていた状態が解消します。



**08** **Blast** ノードを追加して、**Group** を **0** に設定します。これにより、導火線のジオメトリの終端が削除されます。**Normal** ノードをチェーンの終端に追加します。

**normal** ノードの後に **Null** ノードを追加して、**FUSE\_OUT** と名前を付けます。これで、導火線全体のジオメトリを表すノードができました。

オブジェクトレベルに戻り、オブジェクトの名前を **fuse\_geo** に変更します。

## パート3

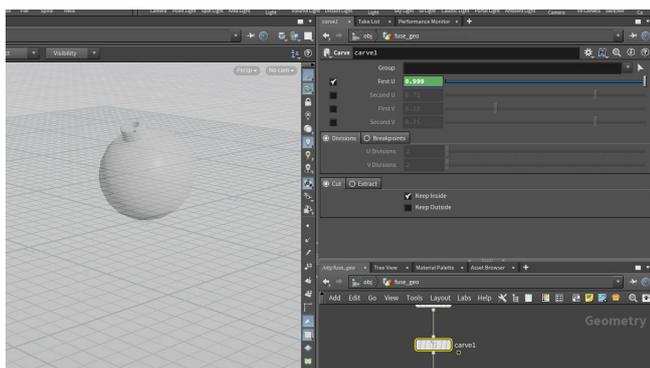
# 導火線のアニメート

導火線のアニメーションでは、Carve ノードを使用して、時間の経過とともにカーブの長さが変化するようにします。導火線に丸いキャップを追加して、すすと火花を放出するのに使用します。カーブ上に接線をセットアップして、キャップが適切に動くようにします。次に、NULL オブジェクトをいくつか追加して、パーティクルの放出に使用するキャップを簡単にエクスポートできるようにします。



**01** fuse\_geo オブジェクトの中に入ります。transform ノードと Polywire ノードの間に、Carve ノードを追加します。First U スライドをドラッグして、カーブにどのような影響を与えるのか確認します。

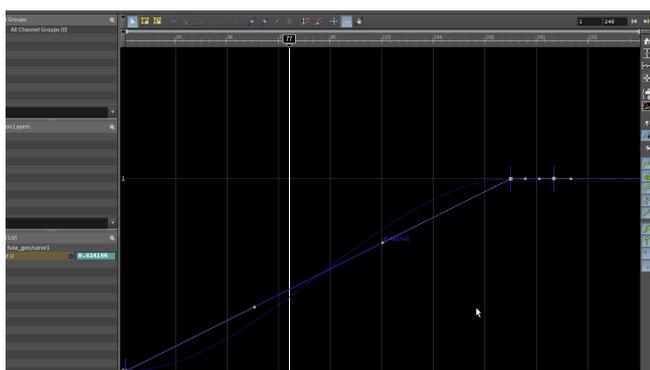
First U を 0 に設定します。First U を Alt クリックして、フレーム 1 にキーフレームを設定します。パラメータボックスの色が変わり、キーフレームが設定されたことと、現在のフレームにキーがあることが示されます。



**02** フレーム 180 に移動します。First U を 0.999 に設定します。これにより、キーフレームが設定されます。フレーム 200 に移動します。First U を 1.0 に設定します。これにより、もう1つキーフレームが設定されます。

プレイヤーの左下で、再生が速くなりすぎないように Real Time Playback をオンに切り替えて、Play を押します。

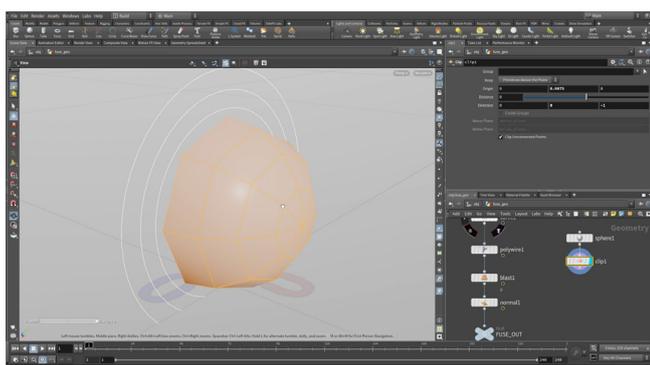
これで、導火線が爆弾のジオメトリに向かってアニメートするようになりました。爆弾には爆発をセットアップします。



**03** Animation Editor ペインタブをクリックします。アニメーションカーブを選択して、パネル上部の Straight ボタンをクリックします。カーブが直線になり、導火線の最初から最後まで、速くなったり遅くなったりせず、均一の速度にアニメートされます。

Scene View ペインタブに戻って、アニメーションを再生し、この変更がモーションに与える影響を確認します。

作業内容を保存します。



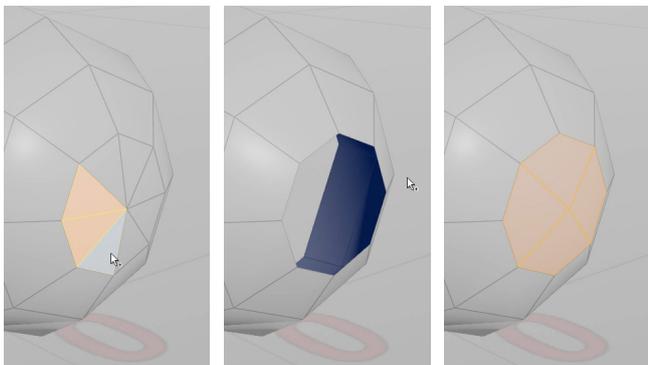
**04** ネットワークビューで、Sphere ノードを追加して、Display フラグを設定します。次のように設定します。

- Radius を 1, 1, 1 にする
- Center Y を 0.0075 にする
- Uniform Scale を 0.0075 にする

スペースバー + F を押して、球にフォーカスします。

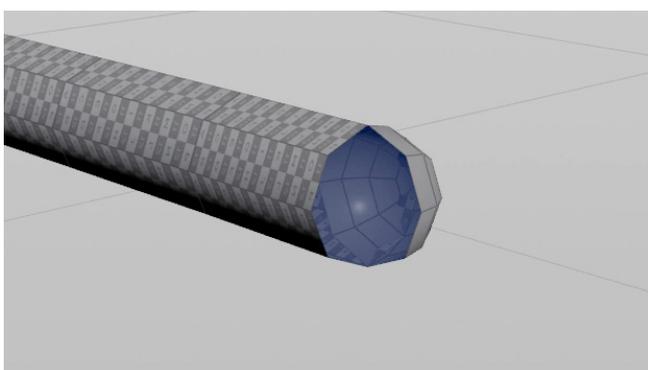
- Orientation を Z Axis にする
- Rows を 9、Columns を 8 にする

Scene View で、N を押してすべてを選択してから、Tab > Clip ノードを選択し、Direction を 0, 0, -1 に設定します。



**05** タンブルして、**S**を押して Select ツールにし、**4**を押してフェース/プリミティブ選択にします。球の先端で三角形の1つを選択したら、**A**キーを押しながら2つの三角形を **MMB** クリックして、三角形の面をすべて選択します。**Delete** キーを押して、それらを削除します。これにより、ネットワークに **Blast** ノードが追加されます。

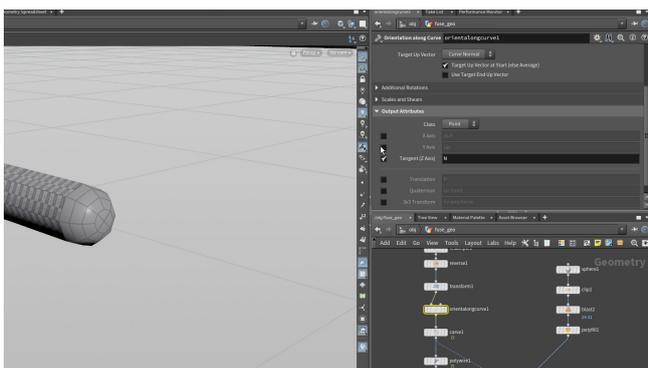
**3**を押してエッジ選択に切り替え、先ほど削除した領域のエッジをダブルクリックします。**Tab > polyfill**を押して、**blast**の後にノードを配置します。**Fill Mode**を **Quadrilateral Grid**に設定し、**Display フラグ**をオンにします。**Smooth**を **100**に、**Tangent Strength**を **0**に設定します。これにより、球の先端に四角形トポロジが作成されます。



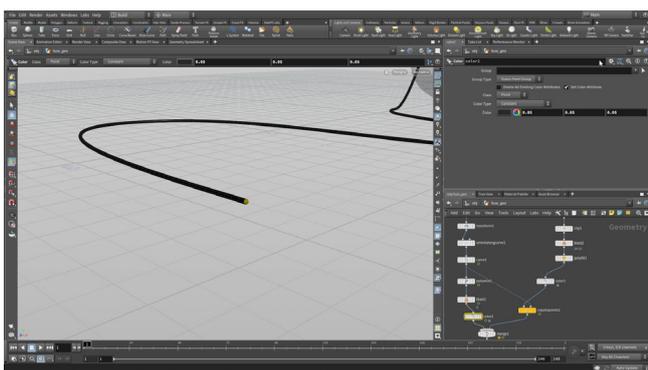
**06** **Copy to Points** ノードをネットワークに追加します。球からの **polyfill** ノードを1つ目の入力に、**carve** ノードを2つ目の入力に接続します。**Target Points**を **0**に設定します。

**Merge** ノードを追加します。**blast** ノードと **copytopoints** ノードを接続してから、**normal** ノードに接続します。

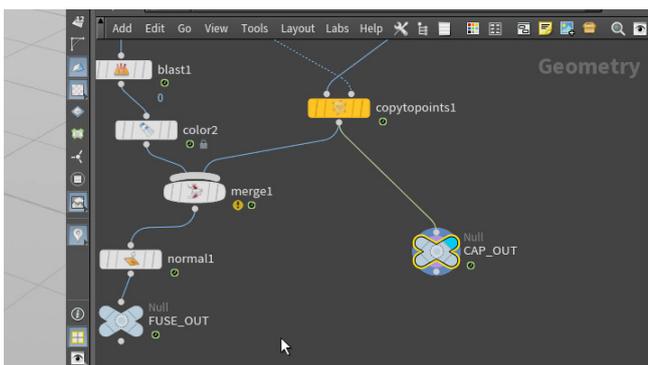
キャップはカーブの終端に適切に配置されましたが、方向が正しくありません。カーブに法線を追加して、適切に揃える必要があります。



**07** **reverse** ノードと **carve** ノードの間に、**Orientation Along Curve** ノードを追加します。**Output Attributes**で **Y Axis** オプションをオフにします。**Tangent (Z Axis)**は **N**のままにしておきます。これにより、法線がカーブに追加され、先端のキャップが導火線に合わせて動くようになります。



**08** **polyfill** ノードの後に **Color** ノードを追加して、Color を黄色に設定します。導火線の **blast** ノードの後に **Color** ノードをもう1つ追加して、Color を **ダークグレー**に設定します。こうした色分けは、作業中に、導火線を視覚化するのに役立ちます。また、これらの色を基に、後で割り当てるマテリアルに影響を与えることも可能です。



**09** **copytopoints** ノードの後で **Null** ノードを分岐させて、横に配置します。**null** ノードの名前を **CAP\_OUT** に変更します。後でこれを使用して、キャップを別のネットワークに抽出します。パーティクルの放出では、そのネットワークを参照することになります。**Display フラグ**を設定し、球の半分が表示されることを確認します。**プレイバー**をスクラップして、球が導火線とともに動くことを確認します。

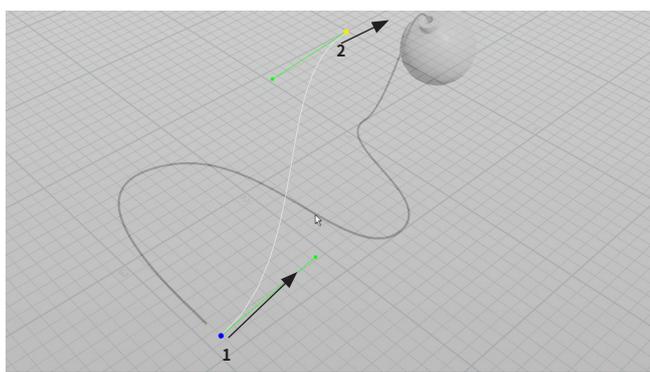
完了したら、**Display フラグ**の設定を **FUSE\_OUT null**に戻します。

## パート4

# アニメーションカメラの作成

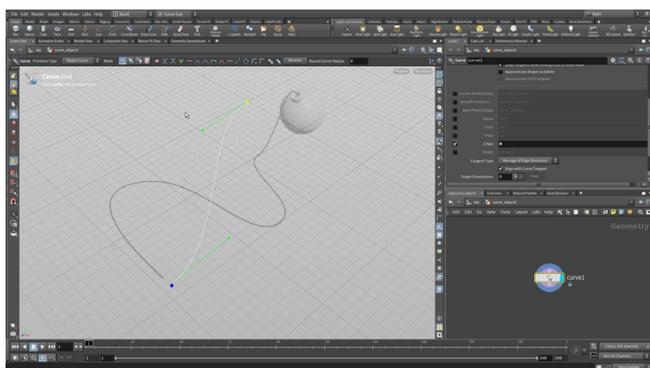
このショットをさらに発展させていくうえで、最終ショットをフレーミングするカメラをセットアップしておく便利です。

このカメラリグを構築するには、Null オブジェクトをカーブに拘束してから、Aim 拘束を使用してカメラを Null オブジェクトに向けます。カメラが導火線の終端に追従するようになるため、パーティクルの放出を評価しやすくなります。

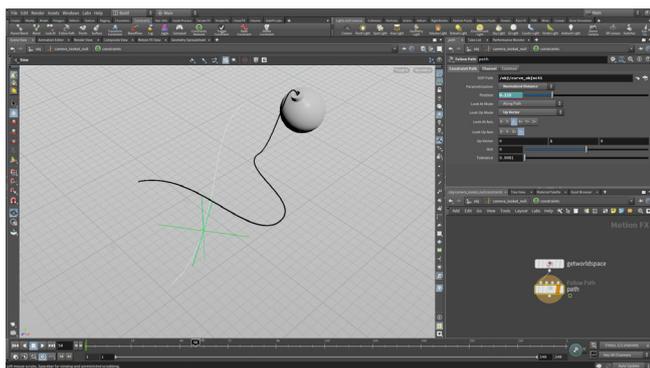


**01** ドリーアウトして上からシーン全体を表示します。**コンストラクション平面**がオンになっていることを確認します。**C**を押して Radial メニューを表示し、**Create > Geometry > Curve** を選択します。導火線の開始付近のポイントを**クリックアンドドラッグ**したら、さらにドラッグして接線を伸ばします。

次に、爆弾の背後で2つ目のポイントを**クリックアンドドラッグ**したら、さらにドラッグして接線を伸ばし、カーブの形状を作成します。**MMB** クリックして終了し、形状を調整したい場合は **Edit** モードを使用します。

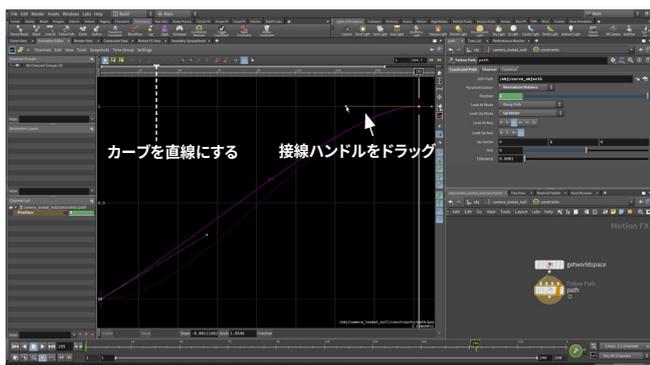


**02** Curve ノードで、Z Axis オプションをオンにして、N に設定します。これにより、カーブに沿ってアニメーションを補助する法線が作成されます。



**03** 原点に Null オブジェクトを追加します。このノードの名前を **camera\_lookat\_null** に変更します。**Constraints** シェルフから、**Follow Path** ツールをクリックします。これで、Null が開始オブジェクトとして受け入れられます。パスオブジェクトとしてカーブを選択し、**Enter** を押します。**look at** オブジェクトまたは **look up** オブジェクトは必要ないので、**Enter** をもう 2 回押します。

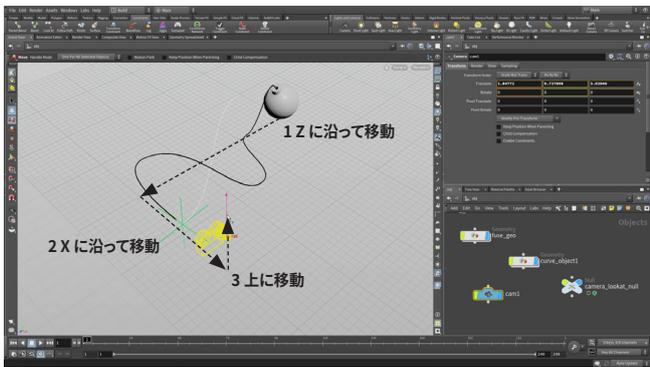
タイムラインをスクラブすると、開始フレームから終了フレームまで、Null オブジェクトがパスに沿って一定のペースで移動することを確認できます。



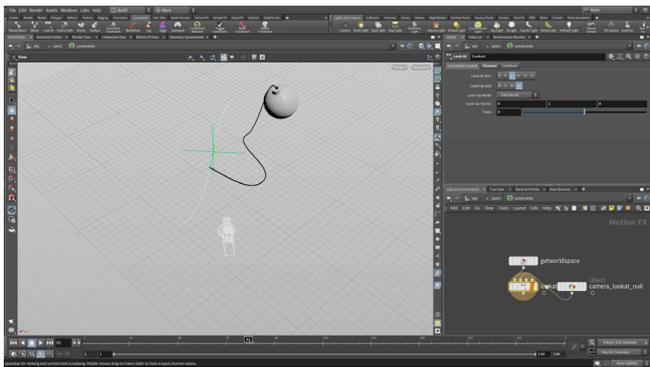
**04** 拘束ネットワークが Null オブジェクトに追加され、path ノードが作成されました。**フレーム 1** に移動します。Path ノードを選択したまま、**Position** パラメータを RMB クリックして **Delete Channels** を選択し、パスに沿って Null をアニメートしているエクスプレッションを削除します。**Position** を **0** に設定します。**Position** を **Alt** クリックして、キーフレームを設定します。

フレーム 195 に移動して、**Position** を **1** に設定します。**Position** を **Alt** クリックして、2 つ目のキーフレームを設定します。

**Animation Editor** タブをクリックし、**H** を押してカーブ全体を表示します。カーブを選択し、**Straight** ボタンを押して直線にします。2 つ目のポイントの接線を使用して、最後が滑らかなように調整します。

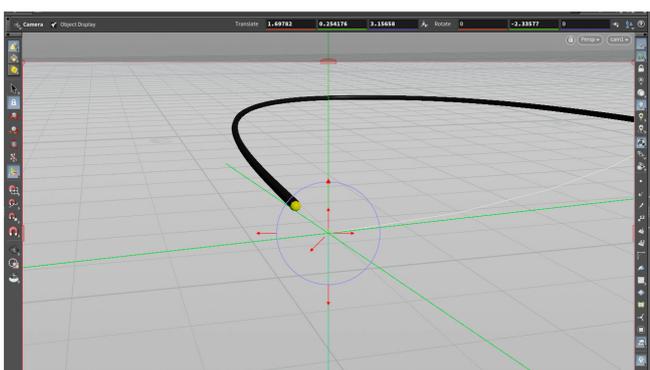


**05** ネットワークビューで、**Tab > Camera** を選択し、**Enter** を押して、クリックして原点に配置します。**Move** ツールを使用して、カメラを導火線の前の少し右側に移動します。Y 軸に沿って上に移動し、地面から約 **0.75** ユニット上げます。



**06** **Constraints** シェルフから、**Look At** ツールをクリックします。これにより、選択したカメラが **look at** オブジェクトとして使用されます。**null** オブジェクトを **look at** オブジェクトとして選択し、**Enter** を押します。再度 **Enter** を押して、**look up** オブジェクトとしては何も割り当てないようにします。

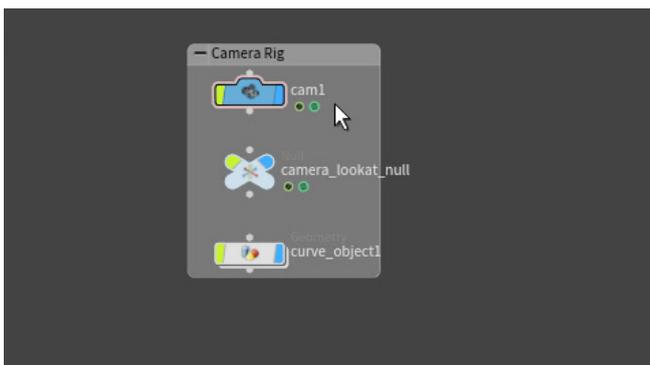
これで、カメラが Null オブジェクトに向くようになりました。Camera メニューから **cam1** を選択し、このカメラ越しに見てみましょう。



**07** **フレーム 1** に移動します。camera を選択して、**Handle** ツールがアクティブなことを確認します。カメラハンドルが表示されるので、ハンドルを使用して導火線の開始点がよく見えるようにカメラを再配置します。

**フレーム 195** に移動します。同じハンドルを使用して、爆弾がフレーム内の適切な位置に見えるようにカメラを再配置します。

シーケンスを通してカメラが適切に動作するように、前後にスクラブして微調整しましょう。シーケンス全体を通して、導火線が見えるようにします。



**08** オブジェクトレベルに移動して、**curve**、**camera\_lookat\_null** オブジェクトと **cam1** ノードを選択して整列させ、ネットワークボックスに入れます。ボックスのタイトルバーをダブルクリックし、ボックスの名前を **Camera Rig** にします。すべてのパーツの Display フラグを**オフ**にして、作業時に Scene View に表示されないようにします。このボックスは必要に応じて折り畳んだり、展開したままにすることができます。

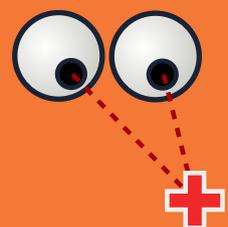
シーンを**保存**します。



## 拘束

Null オブジェクトをパスに追従させ、カメラをパスに向かせるために、**Constraints** シェルフにあるアニメーション拘束を使用しました。これらは、**チャンネルオペレータ**または **CHOP** と呼ばれる特別なノードタイプを使って行われます。これらのノードは、null ノードや camera ノードの内部にあります。これらを使用して、拘束の動作を制御することができます。

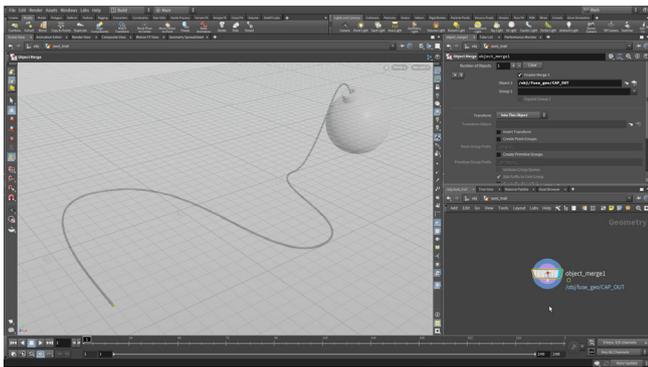
CHOP ノードは、任意のパラメータを **RMB** クリックすると表示される **Motion FX** メニューでも使用できます。



## パート5

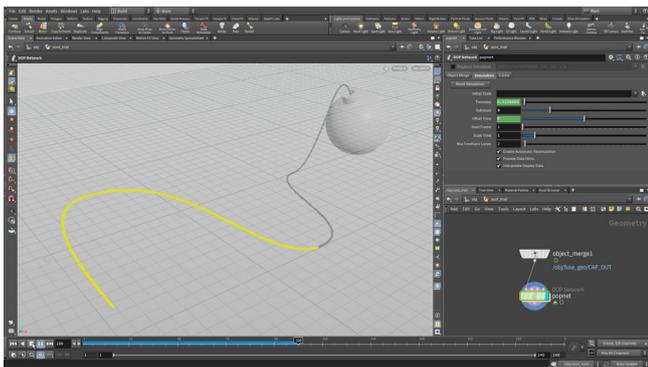
# すすのトレイルの作成

すすのトレイルを作成するには、キャップを使用してパーティクルトレイルを放出します。これらのポイントを適切に放出する方法と、重力などのフォースを追加してパーティクルの動きを制御する方法を学びます。また、衝突をセットアップして、パーティクルが地面にぶつかったり、爆弾のサーフェスを滑り落ちるようにする方法も学びます。



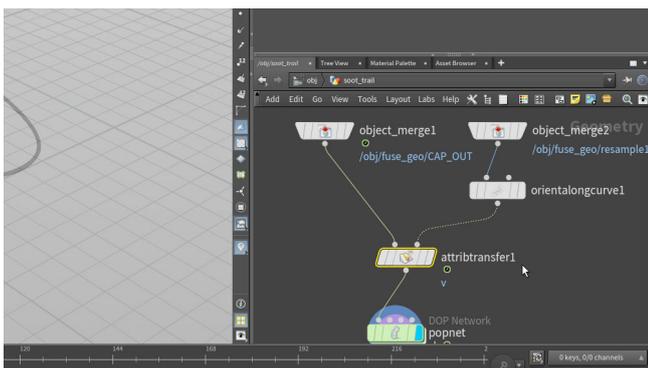
**01** *fuse\_geo* オブジェクトのノードの中に入ります。**CAP\_OUT** の **Display** フラグをオンにした状態で、**Modify** シェルフに移動して、**Extract** ツールを使用します。**N** を押してすべてのフェースを選択し、**Enter** を押すと、Object Merge ノードを使ってキャップを取り込んだ新しいオブジェクトが作成されます。オブジェクトレベルに移動して、このオブジェクトの名前を *soot\_trail* にします。

*fuse\_geo* オブジェクトに戻り、FUSE\_OUT に **Display** フラグを設定します。これで、レンダリング時には導火線のジオメトリ全体が表示され、パーティクルの生成には新しいオブジェクトが使用されます。



**02** *soot\_trail* ノードの中に入り、POP Network ノードをチェーンの終端に追加します。その中に入り、**Source First Input** ノードで、**Const Birth Rate** を **1000** に設定します。**Play** を押します。パーティクルが放出されるのを確認できますが、ただそれだけです。

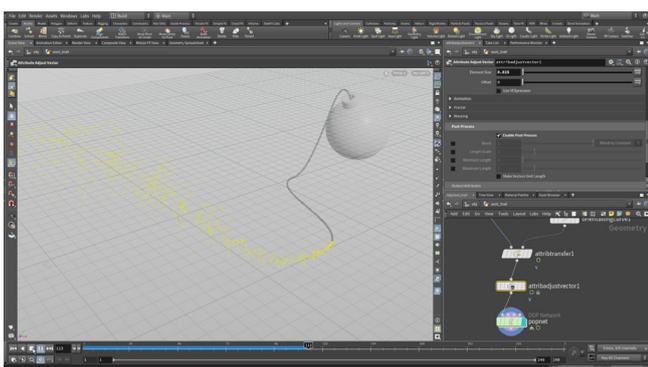
1つ上のレベルに戻り、**Simulation** タブで **Substeps** を **3** に設定します。**Play** を押します。パーティクルがより均一に放出されているのがわかります。



**03** *object\_merge* ノードを **Alt** ドラッグして、2つ目のコピーを作成します。**Object 1** を *fuse\_geo*>*resample* ノードに設定します。このカーブを使用して、キャップに Velocity を転送します。

**Orient Along Curve** ノードを追加したら、**Output Attributes of Tangent (Z Axis)** をオンにして、**v** に設定します。

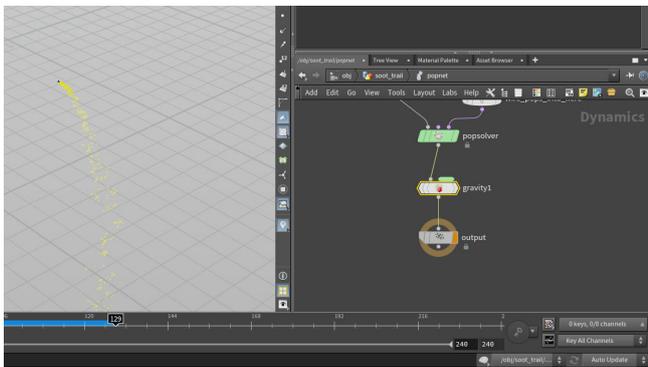
オリジナルの *object\_merge* ノードと *popnet* ノードの間に **Attribute Transfer** ノードを追加します。**Primitives** チェックボックスをオフにして、**Points** を **v** に設定します。



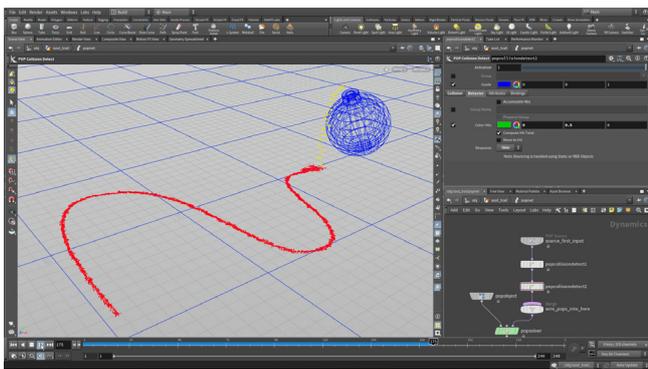
**04** *attribtransfer* と *popnet* の間に **Attribute Adjust Vector** ノードを配置します。Adjustment Value で次のように設定します。

- **Adjustment for** を **Direction Only** にする
- **Adjust with** を **Noise** にする
- **Range Values** を **Zero Centered** にする
- **Amplitude** を **0.5** にする

**Noise Pattern** の **Element Size** を **0.025** に設定し、**Post-Process** の **Enable Post-Process** をオンにします。**Play** を押してテストします。



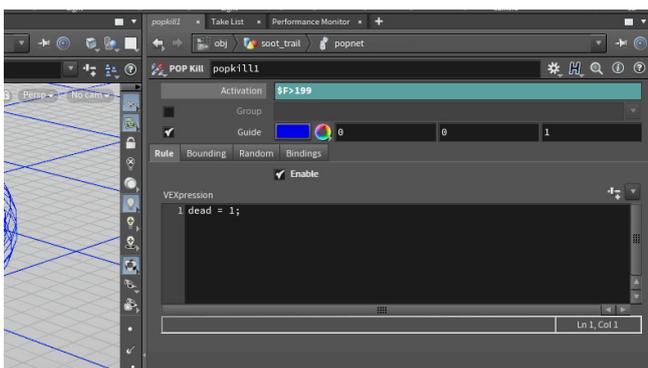
**05** *popnet* の中に入って、*popsolver* の下に **Gravity Force** ノードを追加します。これでシミュレーションを再生すると、パーティクルが地面に落ちようになります。



**06** オブジェクトレベルに戻り、**グリッド**を作成します。**Size** を **30, 30**、**Rows** と **Columns** を **31, 31** にします。

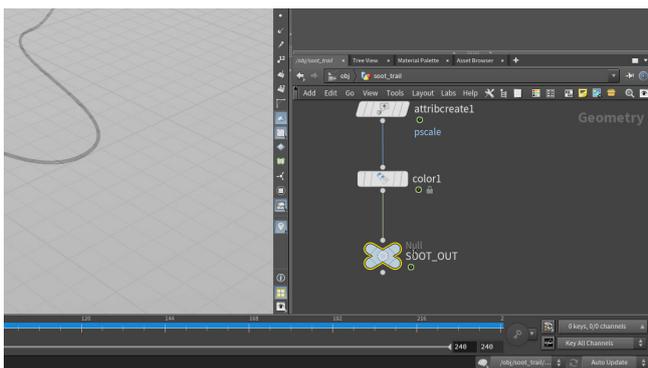
名前を **ground** に変更します。*popnet* の中に入り、**Pop Collision Detect** ノードを **source\_first\_input** ノードの後に追加します。**SOP Path** を **obj/ground/grid1** に設定します。**Behavior** タブで、**Response** を **Stick** に設定します。**Color Hits** は赤色のままにしておきます。

**Pop Collision Detect** ノードをもう1つ追加します。**SOP Path** を **bomb\_geo** ジオメトリオブジェクトに設定します。**Response** を **Slide** に設定します。**Color Hits** を **緑色** に変更します。**Play** を押します。



**07** 現在は、パーティクルがシーケンス全体を通して放出されています。爆弾が爆発する瞬間に、パーティクルを止めなければなりません。**POP Kill** ノードを **wire\_pops\_into\_here merge** ノードの後に追加します。フレーム1に移動して、**Activation** を **\$F>199** に設定します。これで、パーティクルはフレーム200で終了します。

Rule タブで **Enable** をオンにします。**Play** を押してテストします。



**08** ジオメトリコンテキストに戻り、**attrib create** ノードを作成して *popnet* に接続します。**Name** を **pscale** に、**Value** を **0.001** に設定します。

**Color SOP** を追加して、**Color** を **ダークグレー** に設定します。

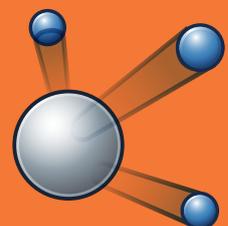
**null** ノードをチェーンの終端に追加して、名前を **SOOT\_OUT** に変更します。



## パーティクル FX

すずのトレイルは、パーティクルダイナミクスを使用して作成します。パーティクルとは、風や重力といったフォースを使用して影響を与えることのできるポイントです。導火線の末端から、ポイントが発生させ、さまざまなテクニックでシミュレートすることができます。

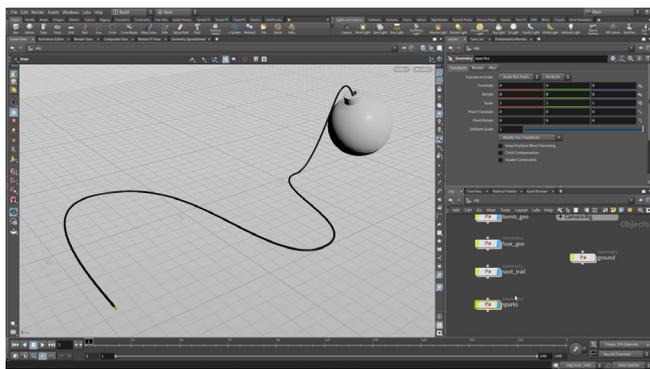
パーティクルは Houdini の Dynamics または DOP セクションを使用してシミュレートした後、SOP に戻してジオメトリとして扱うことができます。次のセクションでは、パーティクルを使用して導火線の末端に火花を作成します。



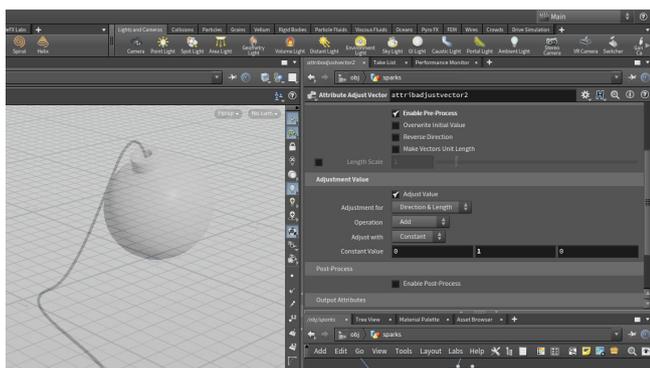
## パート 6

# パーティクルの火花の作成

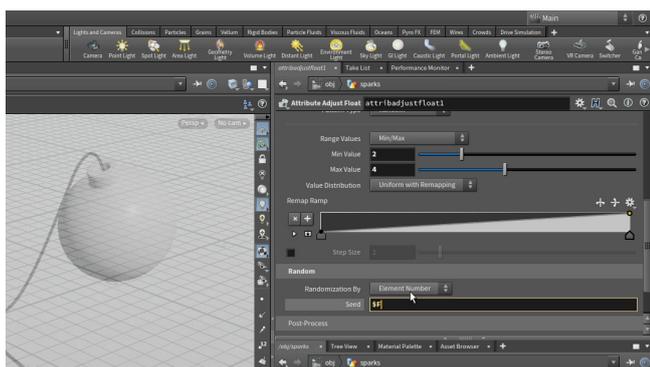
火花を作成するには、すずのパーティクルオブジェクトをコピーして、火花を生成する新しいオブジェクトとなるように設定を変更します。火花のパーティクルの方が寿命が短く、活発になるようにします。Spark Trail ノードを使用すると、希望通りのルックの火花をショットに追加することができます。このノードのパラメータを調整することで、目的のルックを実現します。



**01** フレーム 1 に戻り、オブジェクトレベルに移動します。Ground オブジェクトを非表示にします。soot\_trail オブジェクトを Alt ドラッグして、コピーを作成します。コピーの名前を sparks にします。この新しいオブジェクトには既に popnet があり、これを火花のパーティクルシミュレーションを生成するように変更します。Houdini では、すべてをゼロから作成するのではなく、既にあるネットワークを再利用することを勧めます。



**02** sparks オブジェクトに入ります。いくつか変更を加えて、火花を作成するネットワークをセットアップします。attributecreate と color ノードを削除します。これらはこのパーティクルネットワークには必要ありません。null ノードの名前を SPARKS\_OUT に変更します。attributeadjustvector ノードで、Amplitude を 1.75 に変更します。このノードの上に、新しい Attribute Adjust Vector ノードを追加します。Enable Pre-Process をオンにして、Constant Value を 0, 1, 0 に設定します。これで、パーティクルが上昇してから落下するようになりました。Play を押します。



**03** フレーム 1 に戻ります。Attribute Adjust Float ノードを popnet ノードの直前に追加します。次のように設定します。

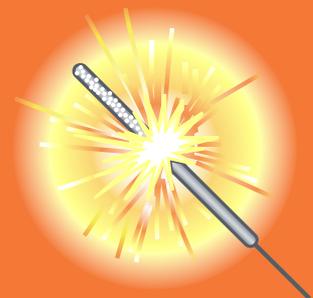
- Attribute Name を life にする
- Unit Settings を Duration にする
- Pattern Type を Random にする
- Min Value を 2、Max Value を 4 にする
- Random で、Seed を \$F にする

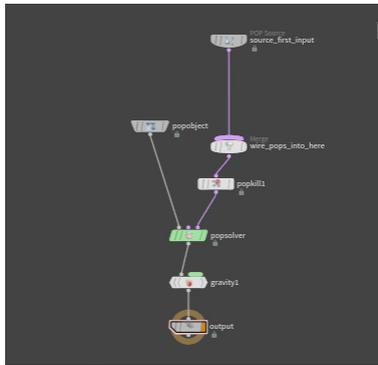
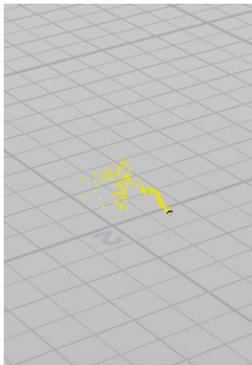


## 火花 | PARTICLE TRAIL

Particle Trail SOP は、アニメーションするパーティクルシステムを受け取り、そのパーティクルからモーショントレイルを生成します。これらのトレイルは、火花、花火、雨などの様々なエフェクトで使用可能です。

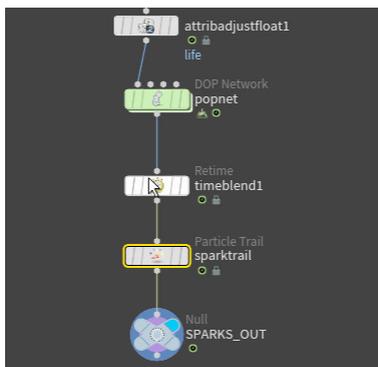
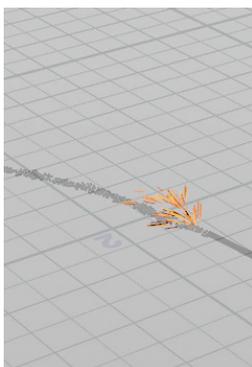
このノードは、そのトレイルのルックを制御することもできます。これによって、長い期間のモーションブラーを使ってポイントをレンダリングすることなく、SOP コンテキスト内でルックを微調整することができます。





**04** *popnet* の中に入り、2つの *collisiondetect* ノードを削除します。この時点では衝突を気にする必要はありません。後から必要に応じて別のノードを追加します。

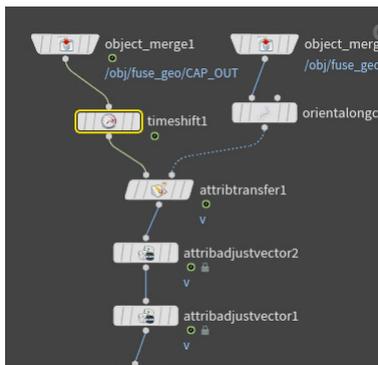
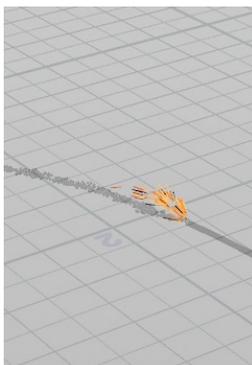
**Play** を押して、パーティクルが放出されるのを確認します。



**05** ジオメトリレベルに戻ります。フレーム 1 に移動します。

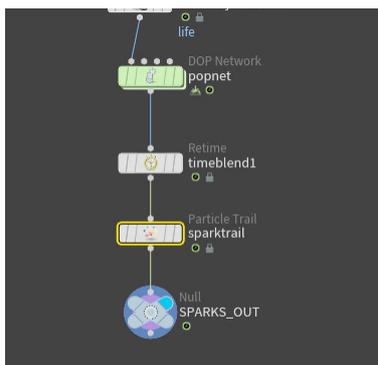
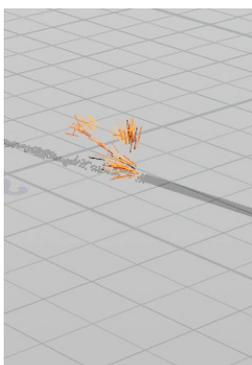
ネットワークビューで、**Tab > Spark Trail** を押します。少し遅れて、*timeblend* と *sparktrail* の2つのノードが追加されます。*popnet* を *timeblend* ノードに接続し、*sparktrail* を *SPARKS\_OUT* Null に接続します。

**Play** を押します。火花が見えますが、導火線の末端よりも遅れています。



**06** キャップを取り込む *object\_merge* と、*attributetransfer* ノードの間に *Time Shift* ノードを追加します。**Frame** パラメータ名をクリックして、使用されているエクスプレッションを確認します。それを  $\$FF + 1$  に変更します。

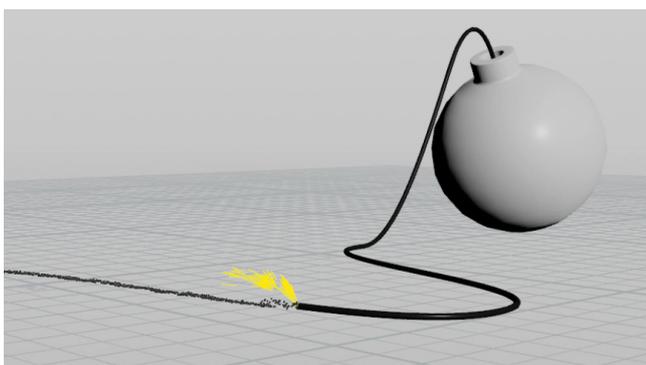
これによりキャップが1フレーム分進むため、火花と導火線の末端が揃うようになります。**Play** を押してテストします。



**07** フレーム 1 に移動し、*sparktrail* ノードを選択します。**Split** タブをクリックし、**Enable Split** チェックボックスをオンにします。**Percent to Split** を **40** に設定します。

**Shape** の **Splits per Point** を **4** に設定します。

**Play** を押してテストします。



**08** フレーム 1 に戻り、オブジェクトレベルに移動します。

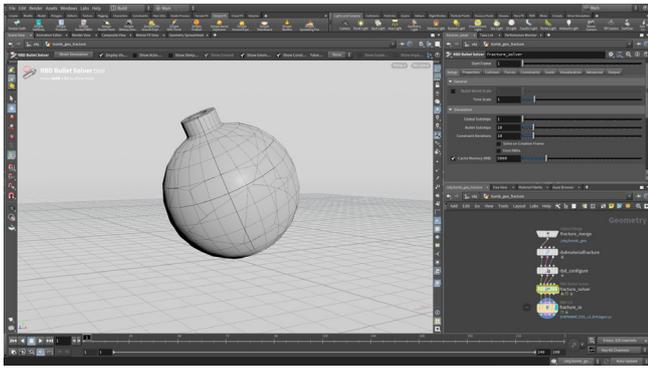
カメラを *cam1* に設定し、ツールバー下部の **Flipbook** ボタンを押します。表示されるウィンドウで **Start** をクリックします。Scene View のアニメーションシーケンスが作成され、パーティクルの動きを評価できます。

**Realtime トグル** がオンになっていることを確認したら、**Flipbook** を再生して、ショットの進行を確認します。

## パート7

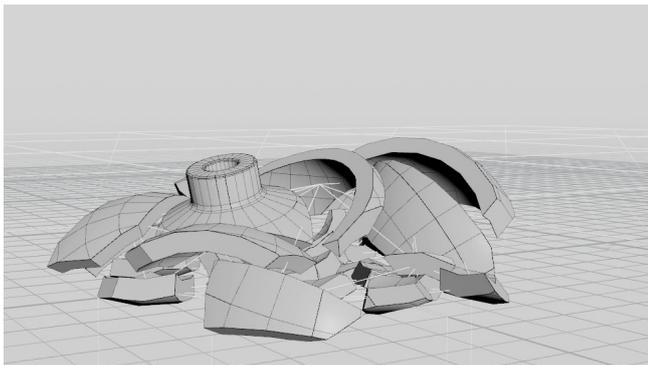
# 爆弾の爆発

爆弾のジオメトリに必要なのは、リジッドボディダイナミクスシミュレーションです。ジオメトリを粉碎してから、アトリビュートを追加して爆発を作成します。その後、移動するパーツの速度を制御して、芸術的な観点でルックを調整します。シミュレーションの準備が整ったら、ジオメトリをキャッシュして、効率的に PyroFX の段階に進めるようにします。



**01** *fuse\_geo*、*soot\_trail*、*sparks* オブジェクトを非表示にします。スペースバー + G を押して、爆弾のジオメトリを中心に寄せます。プレイヤーでフレーム 1 に移動します。スペースバー + G を押して、爆弾を中心に寄せます。

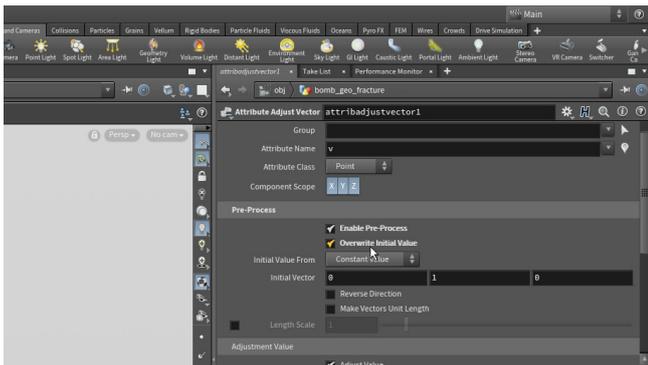
*bomb\_geo* オブジェクトを選択し、Simple FX シェルフから Simple Fracture を選択します。衝突オブジェクトの選択を求められたら、何も選択せずに Enter を押します。すると、このネットワークがすぐにセットアップされます。このオブジェクトは爆弾のジオメトリを結合して、粉碎とシミュレーション用のノードをセットアップします。



**02** *fracture\_solver* ノードで、Start Frame を 200 に設定します。プレイヤーの開始フレームを 200 に設定します。First Frame ボタンをクリックし、フレーム 200 に移動します。爆発のシミュレーションが必要なのはフレーム 200 から 240 のみです。

*fracture\_solver* ノードで Collision タブに移動して、Ground Collision セクションの Ground Type を Ground Plane に設定します。Advanced タブに移動し、Constraints > Glue セクションで、Data Name フィールドの Glue という語を削除します。

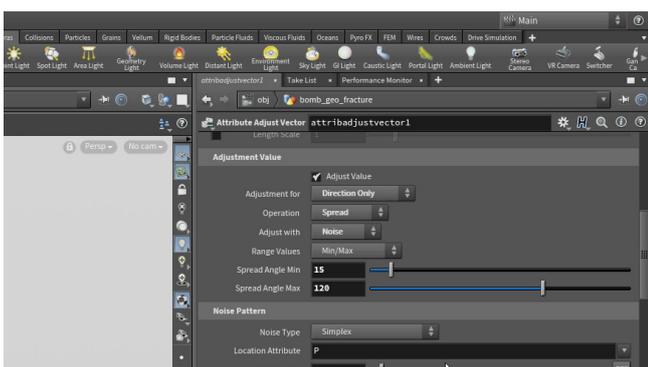
Play を押して、爆弾がバラバラになるのを確認します。次は、パーツを爆発させるために開始 Velocity を追加します。



**03** フレーム 200 に移動します。Attribute Adjust Vector ノードを追加して、他のノードの右側に配置します。*rbd\_configure* ノードの 3 目目の出力を *attributeadjustvector* ノードに接続し、*attributeadjustvector* ノードを *fracture\_solver* ノードの 3 目目の入力に接続します。

このノードで次のように設定します。

- Enable Pre-Process をオンにする
- Overwrite Initial Value をオンにする
- Initial Vector を 0, 1, 0 にする



**04** Adjustment Value で、次のように設定します。

- Adjustment for を Direction Only にする
- Operation を Spread にする
- Adjust with を Noise にする
- Spread Angle Min を 15 にする
- Spread Angle Max を 120 にする

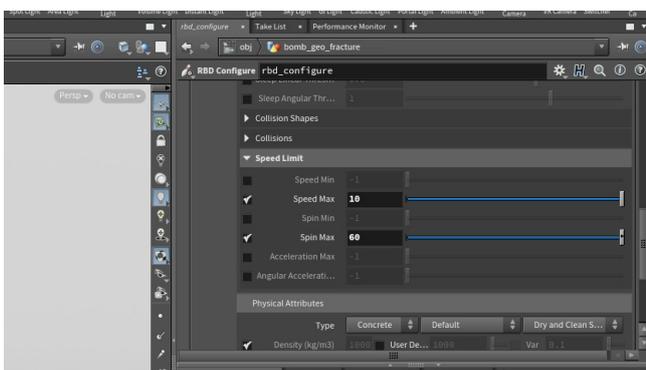
Noise Pattern で Element Size を 0.5 に設定します。

Post-Process で Enable Post-Process をオンにしたら、Minimum Length をオンにして、20 に設定します。



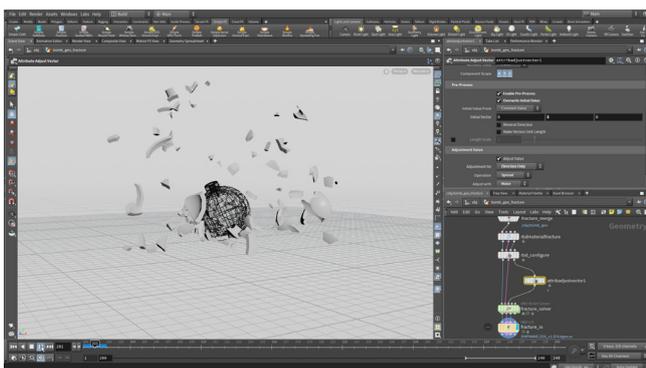
**05** **Play** を押します。爆弾が爆発するようになりました。ジオメトリに対する Velocity アトリビュートの設定がシミュレーションに送られ、破片の初期 Velocity を使用して破片を押し動かします。

Houdini では、多くのビジュアルエフェクトのセットアップで、アトリビュートの操作が効果的です。



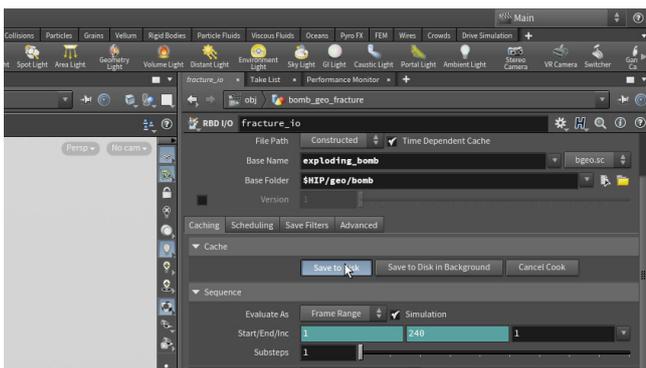
**06** **rbdconfigure** ノードを選択します。Speed Limit セクションを開き、Speed Max と Spin Max パラメータをオンにします。Speed Max を 2、Spin Max を 30 に設定します。Play を押します。シミュレーションが元の速度よりもだいぶ遅くなります。

今度は、Speed Max を 10、Spin Max を 60 に設定します。Play を押します。

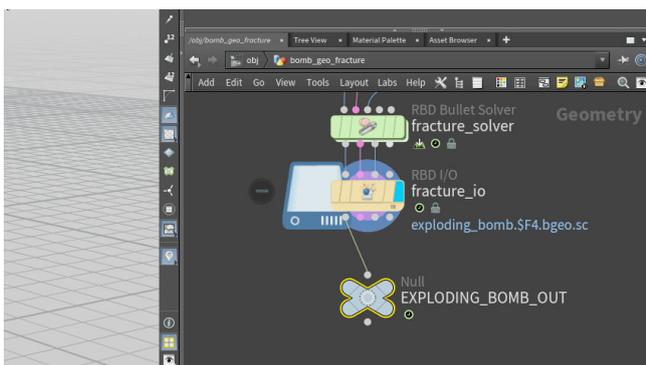


**07** **rbdmaterialfracture** ノードを選択して、Cell Points の Scatter Points を 25 に設定します。Play を押します。これで、破片がかなり増えました。

**attributeadjustvector** ノードに戻り、Length Scale を 50、Initial Vector を 0, 5, 0 に設定します。Play を押します。



**08** フレーム 200 に移動します。fracture\_io ノードで、Base Name を exploding\_bomb、Base Folder を \$HIP/geo/bomb に設定します。Save to Disk をクリックすると、Load From Disk がオンになります。Play を押して、キャッシュ化されたジオメトリをプレビューします。

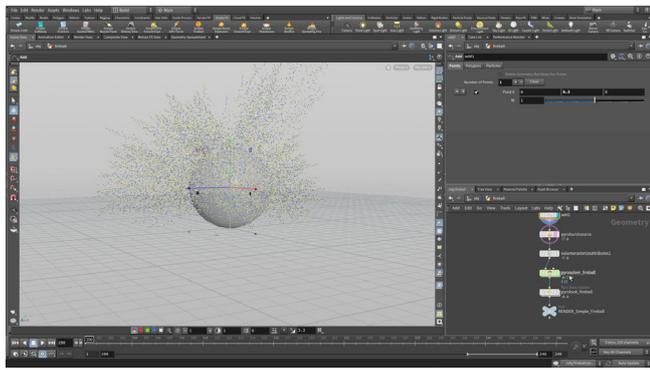


**09** Null ノードを追加して、それに fracture\_io ノードの 1 つ目の出力を接続します。このノードを EXPLODING\_BOMB\_OUT という名前に変更します。

## パート 8

# PyroFX の爆発の作成

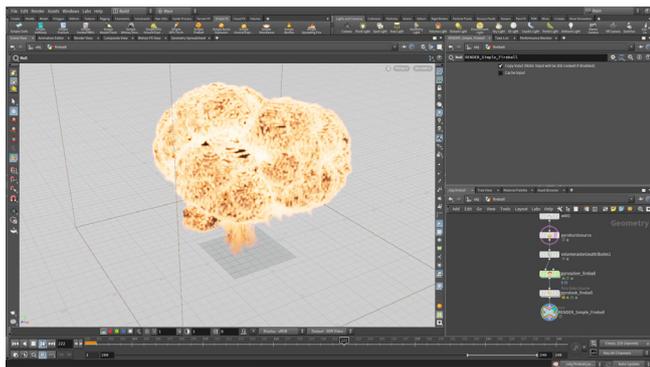
爆弾の爆発には、火の玉が伴います。GPU で動作する Simple Fireball から始め、変更を加えてショットにふさわしいリックを作成します。また、爆発する爆弾のパーツを組み込んで、PyroFX ボリュームを押ししたり、作用したり、さらに面白い見た目にもすることもできます。



**01** オブジェクトレベルに移動します。Simple FX シェルフから、Simple Fireball をクリックします。Scene View で、Enter を押して原点に配置します。これにより、fireball オブジェクトの内部に複数のノードが作成されます。

pyroburstsource ノードに Display フラグを設定し、Burst Animation タブで Start Frame を 200 に設定します。フレーム 200 に移動します。このノードは、爆発の最初の爆風です。

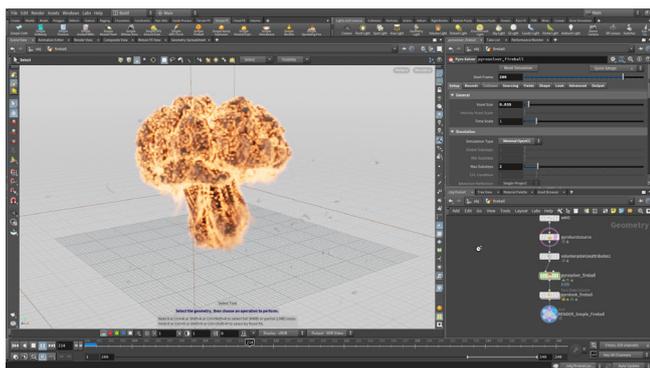
Quick Setups メニューから、Single Input Point を選択します。これにより、1 個のポイントが追加されます。このトランスフォームハンドルを使用して、ポイントを約 0.3 引き上げ、爆弾のジオメトリの中央に配置します。



**02** pyrosolver\_fireball ノードで、Start Frame を 200 に変更します。Simulation Type を Minimal OpenCL に設定し、GPU を使用してシミュレートするようにします。

Sourcing タブで、Limit Source Range オプションを開き、Frame Range を 200, 240 に設定します。Cycle Length をオフにします。

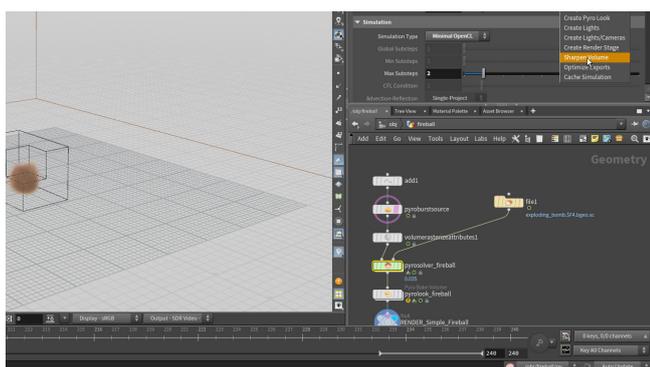
RENDER\_Simple\_Fireball Null ノードに Display フラグを設定します。ズームアウトしてシーンを広く見えるようにしたら、Play を押して、シミュレーションをテストします。ずいぶん大きい火の玉です。さらにズームアウトして爆発全体を表示します。



**03** フレーム 200 に移動します。pyroburstsource ノードで、Burst Shape タブに移動して Initial Size を 0.35、Spread Angle を 180 に設定します。

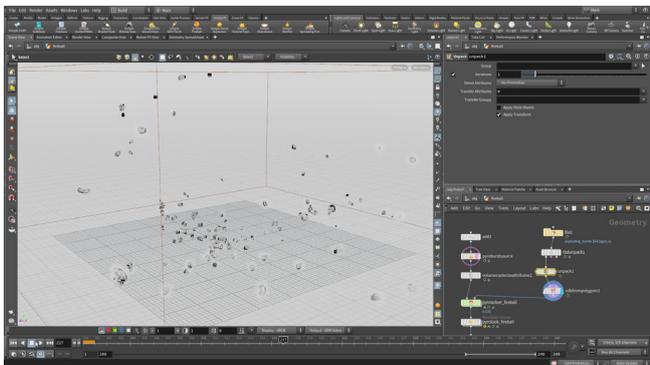
次に fireball の Pyro Solver ノードに移動して Bound タブで Size を 15, 12, 12、Center を 0, 4, 0 に設定します。ボックスが小さくなり、GPU 上で効率的にシミュレートできるようになります。Setup タブで、Voxel Size を 0.035 に設定します。これで、小さくしたシミュレーションにディテールが追加されます。

Play を押して、シミュレーションをテストします。爆発がシーンにうまく収まるようになりました。



**04** 次に、Pyro FX と爆発する爆弾を統合します。File ノードを追加して、ディスクから \$HIP/geo/bomb/exploding\_bomb\_\$.fgeo.sc ジオメトリシーケンスをロードします。このノードを fireball の Pyro Solver ノードの 2 つ目の入力に接続します。

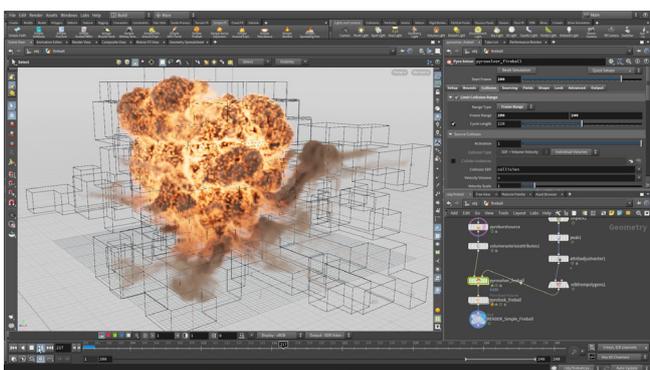
Play を押して、シミュレーションをテストします。適切に準備されていないので、衝突ジオメトリにエフェクトがありません。



**05** フレーム 200 に移動します。**fireball** ノードを選択します。右上の **Quick Setups** メニューから、**Setup SDF Collision** を選択します。これにより、**vdbfrompolygons** ノードが追加されます。このノードの **Voxel Size** は Pyro Solver ノードから取得されます。

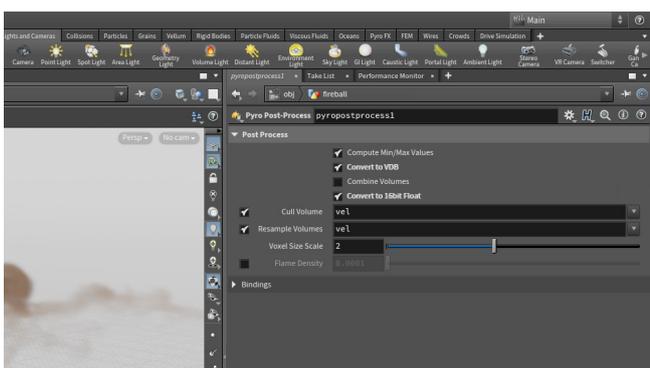
**file** ノードと **vdbfrompolygons** ノードの間に **RBD Unpack** ノードを追加します。

**rbdunpack** ノードと **vdbfrompolygons** ノードの間に **Unpack** ノードを追加します。**unpack** ノードで、**Transfer Velocity** を **v** に設定します。**vdbfrompolygons** ノードに **Display フラグ** を設定し、**Play** を押して衝突ジオメトリを表示します。



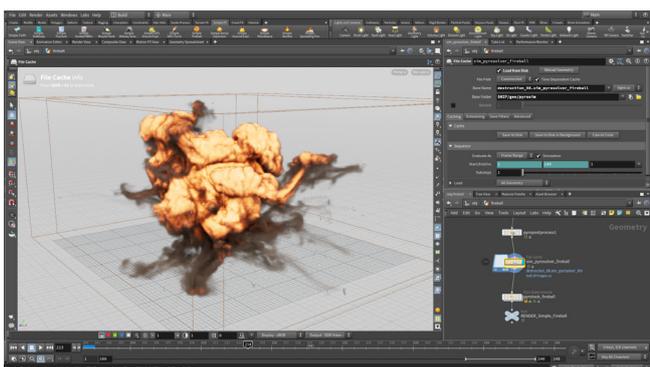
**06** **unpack** と **vdbfrompolygons** の間に、**Peak** ノードを追加して破片を大きくします。**Distance** を **0.1** に設定します。**peak** ノードと **vdbfrompolygons** ノードの間に、**Attribute Adjust Vector** ノードを追加します。**Adjust Value** をオフ、**Enable Post Process** をオンにしてから、**Length Scale** をオンにして **2** に設定します。

**fireball** ノードに **Display フラグ** を再度設定します。**Collision** タブから、**Limit Collision Range** セクションを開き、**Range Type** を **Frame Range**、**Frame Range** を **200, 240** に設定します。**Cycle Length** をオフにします。**Play** を押して、シミュレーションをテストします。爆発する爆弾にコリジョンが構築され、Pyro FX シミュレーションに影響するようになりました。



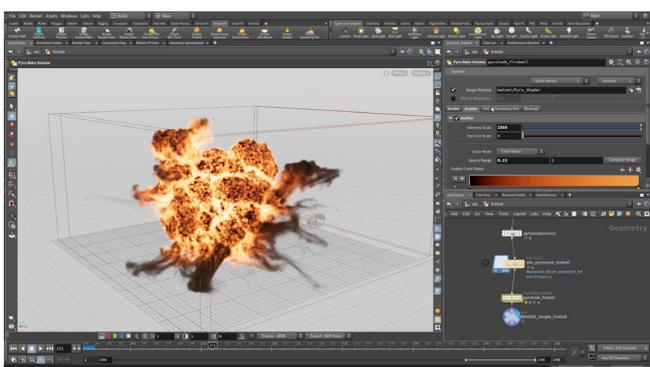
**07** **fireball** の Pyro Solver ノードと **pyrolook** ノードの間に、**Pyro Post Process** ノードを挿入します。**Convert to VDB** と **Convert to 16bit Float** チェックボックスをオンにします。次に、**Cull Volume** と **Resample Volumes** オプションをオンにして、どちらも **vel** のままにしておきます。

このノードにより、ボリウムがより効率的になり、ボリウムをキャッシュ化する際のディスク容量を節約できます。



**08** **fireball** の Pyro Solver ノードで、**Quick Setups** をクリックし、**Cache Simulation** を選択します。そのノードを横に移動して、**pyropostprocess** ノードを **sim\_pyrosolver\_fireball** File Cache ノードに接続し、cache ノードを **pyrolook** ノードに接続します。Base Folder を **\$HIP/geo/pyrosim/** に設定し、**Save to Disk** をクリックします。

**Load from Disk** オプションをオンにしたまま、プレイバーをスクラップして PyroFX の爆発を確認します。



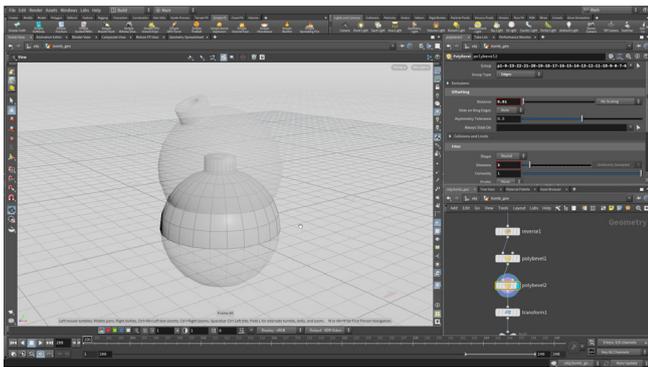
**09** **pyrolook** ノードに **Display フラグ** を設定します。これは **Pyro Bake Volume** ノードで、Scene View でシミュレーションを視覚化するために使用できます。後でレンダリング時に使用する、Pyro Shader とインターフェースがよく似ています。

**Smoke** タブで **Smoke Color** を暗くして、**Scatter** タブで **Intensity Scale** を **2500** に設定します。他の調整も行って、シミュレーションが目的のルックになるようにします。

## パート9

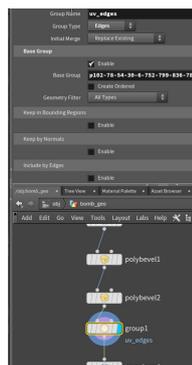
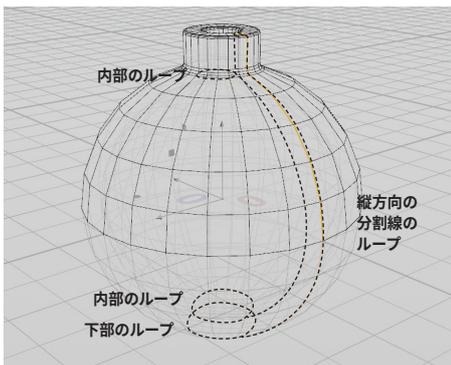
# ジオメトリを USD にエクスポート

レンダリング向けにショットをセットアップするには、USD ファイルにジオメトリをエクスポートして、Solaris コンテキストで参照できるようにする必要があります。ジオメトリを直接インポートすることもできますが、USD としてキャッシュ化すると、シーケンスを固定して、Solaris でライティングとレンダリングに集中できます。これらのオブジェクトの一部では、エクスポートの前に UV を追加して、テクスチャリングに備えます。



**01** オブジェクトレベルに移動して、**fireball** オブジェクトを非表示にします。**bomb\_geo** をダブルクリックし、2つ目の **polybevel** ノードに **Display フラグ**を設定します。これにより、所定の位置に回転される前の状態の爆弾のジオメトリが表示されます。

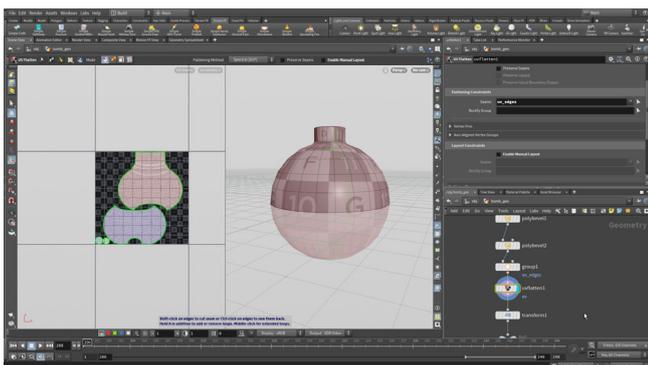
**Hide Other Objects** を設定し、このネットワークのコンテンツに集中できるようにします。



**02** **Select** ツールに移動し、**3** を押してエッジ選択にします。**W** を押してワイヤーフレームモードにしたら、**Shift** を押しながらダブルクリックして下部のループの部分を選択します。**Shift** を押しながらダブルクリックして、4つのすべての領域を選択します。内部ループでも繰り返します。

内部の球の上部にあるループに移動し、**Shift** を押しながらダブルクリックして選択します。X 軸と平行な横方向の分割線を、**Shift** を押しながらダブルクリックして選択します。

Scene View で、**Tab > Group** を押します。**Group Name** を **uv\_edges** に変更します。



**03** Scene View で、**Tab > UV Flatten** を押すと、group ノードの後に **uvflatten** ノードが追加されます。**Seams** を **uv\_edges** に設定し、**Layout Constraints** で **Enable Manual Layout** をオフにします。こうすると UV レイアウトがよく見えるため、爆弾のテクスチャリングに好都合です。

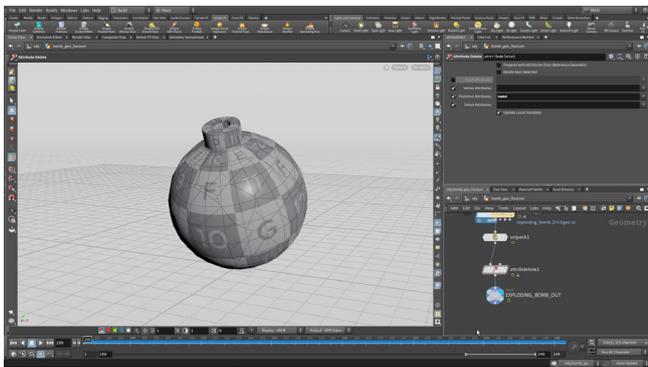
**BOMB\_OUT** null ノードに **Display フラグ**を設定します。



**04** チェーンの終端に **USD Export** ノードを追加します。**static\_bomb** という名前に変更します。次のように設定します。

- **Valid Frame Range** を **Render Current Frame** にする
- **Output File** を **\$HIP/USD/static\_bomb.usd** にする

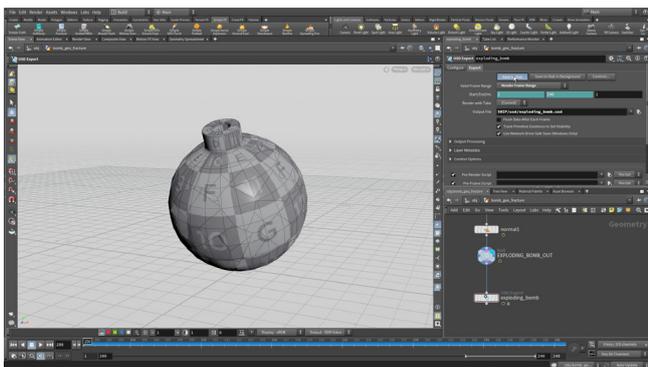
**Save to Disk** を押します。



**05** オブジェクトレベルに戻り、**bomb\_geo\_fracture** ネットワークをダブルクリックします。**fracture\_io** ノードを選択し、**Save to Disk** をクリックして、新しい UV 付きでジオメトリをキャッシュ化します。ジオメトリはバック化されているため表示されません。

**fracture\_io** ノードと **EXPLODING\_BOMB\_OUT** ノードの間に **Unpack** ノードを追加したら、**Attribute Delete** ノードを追加します。**attributedelete** ノードで、**Primitive Attributes** を **name** に設定します。**Point Attributes** セクションをオフにします。

これにより、シーケンスが単一のメッシュとして Solaris に取り込まれます。name アトリビュートはシーケンスを個々のパーツに分割します。

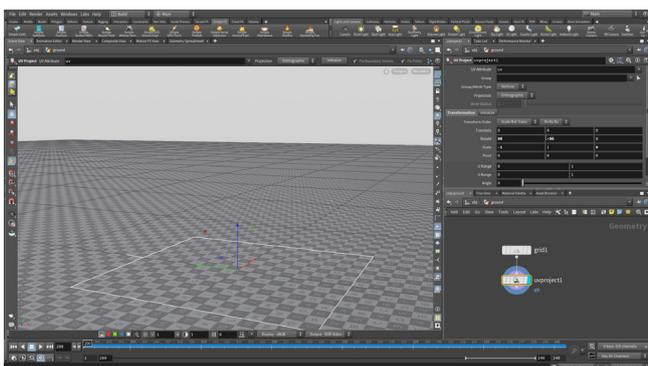


**06** **attributedelete** ノードの後に、**Normal** ノードを追加します。これにより、爆弾のジオメトリが Solaris で適切に表示されるようになります。

チェーンの終端に **USD Export** ノードを追加します。**exploding\_bomb** という名前に変更します。次のように設定します。

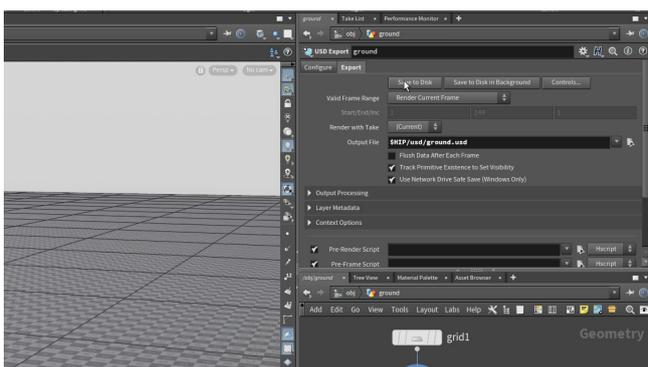
- **Valid Frame Range** を **Render Frame Range** にする
- **Output File** を **\$HIP/usd/exploding\_bomb.usd** にする

**Save to Disk** を押します。



**07** オブジェクトレベルに戻り、**ground** ネットワークをダブルクリックします。**grid** ノードの後に、**UV Project** ノードを追加して、**Display フラグ**を設定します。**Initialize** タブに移動して、**Initialize** ボタンを押します。**Transformation** タブで、**Scale X** を **-1**、**Scale Y** を **1**、**Rotate Y** を **-90** に変更します。

これにより、1つの大きいテクスチャが作成されるのではなく、テクスチャが地面で繰り返されるようになります。



**08** チェーンの終端に **USD Export** ノードを追加します。それを **ground** という名前に変更します。次のように設定します。

- **Valid Frame Range** は **Render Current Frame** のままにする
- **Output File** を **\$HIP/usd/ground.usd** にする

**Save to Disk** を押します。

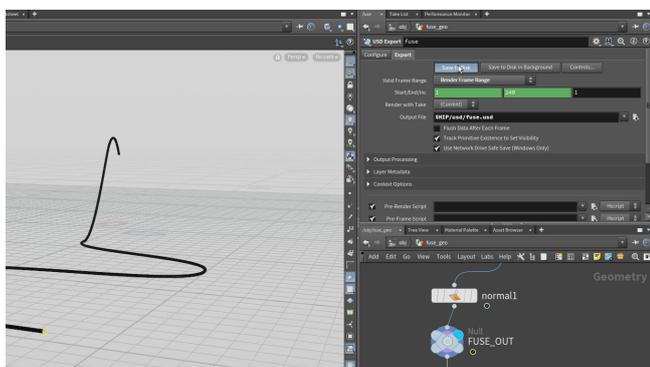


## USD と SOLARIS

このプロジェクトのルックデベロップメントをサポートするために、レイアウト、ルックデブ、ライティングワークフローは **Solaris** コンテキストでセットアップします。これは、**LOP ネットワーク** で表現されます。ここで作成する **USD** キャッシュを、Solaris コンテキストに取り込みます。

このシーンファイルを使用して USD キャッシュを LOP ネットワークで参照しますが、より大規模なパイプラインでは、新しいシーンファイルを開始し、新しいシーンで USD ファイルをインポートする方法もあります。こうするとショットのライティングとレンダリングに集中できますが、ジオメトリとシミュレーションに戻って微調整するのは難しくなります。

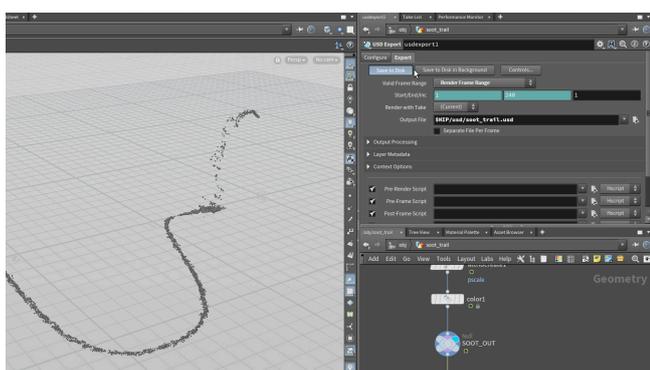




**09** オブジェクトレベルに戻り、**fuse** ネットワークを**ダブルクリック** します。チェーンの終端に **USD Export** ノードを追加します。それを **fuse** という**名前に変更**します。次のように設定します。

- **Valid Frame Range** を **Render Frame Range** にする
- **Output File** を **\$HIP/USD/fuse.usd** にする

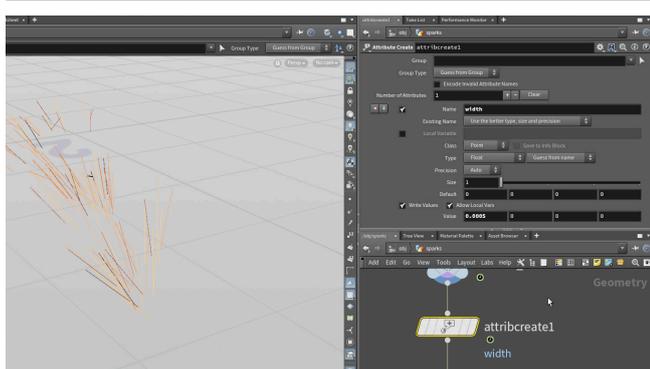
**Save to Disk** を押します。



**10** オブジェクトレベルに戻り、**soot** ネットワークを**ダブルクリック** します。チェーンの終端に **USD Export** ノードを追加します。それを **soot\_trail** という**名前に変更**します。次のように設定します。

- **Valid Frame Range** を **Render Frame Range** にする
- **Output File** を **\$HIP/USD/soot\_trail.usd** にする

**Save to Disk** を押します。



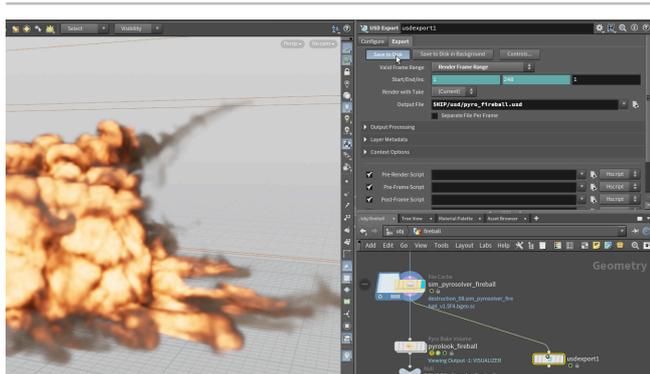
**11** オブジェクトレベルに戻り、**sparks** ネットワークを**ダブルクリック** します。

チェーンの終端に **Attribute Create** ノードを追加します。Name を **width**。Value を **0.0005** に設定します。これにより、レンダリングされる火花のルックが決まります。

チェーンの終端に **USD Export** ノードを追加します。それを **sparks** という**名前に変更**します。次のように設定します。

- **Valid Frame Range** を **Render Frame Range** にする
- **Output File** を **\$HIP/USD/sparks.usd** にする

**Save to Disk** を押します。



**12** オブジェクトレベルに戻り、**fireball** ネットワークを**ダブルクリック** します。**sim\_fireball** ノードから、**USD Export** ノードを分岐させます。**pyrolook** ノードをバイパスしてください。それを **pyro\_fireball** という**名前に変更**します。次のように設定します。

- **Valid Frame Range** を **Render Frame Range** にする
- **Output File** を **\$HIP/USD/pyro\_fireball.usd** にする

**Save to Disk** を押します。



## Scene Import

ジオメトリとシミュレーションを Solaris に取り込むもう 1 つの方法は、**Scene Import LOP** を使用することです。これにより、作業しているジオメトリおよびオブジェクトと、LOP ネットワークの間に、直接的なつながりができます。この方法では、モーションプレーヤーをサポートする Cache LOP が必要になります。この追加のノードは、USD ファイルを参照する場合には必要ありません。

このツールは、アニメーションカメラをオブジェクトレベルから Solaris コンテキストに取り込むのに使用します。



## パート 10

# Solaris でのショットのセットアップ

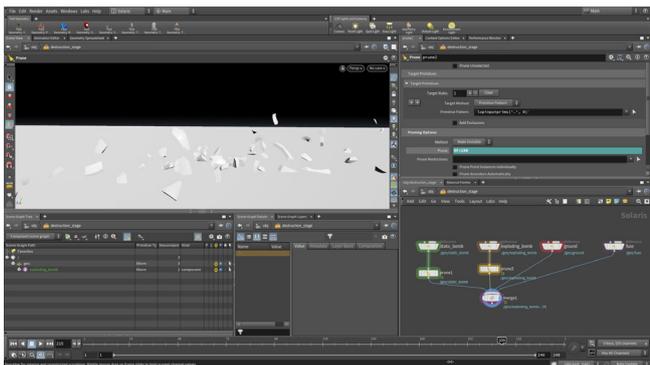
すべての USD ファイルを Solaris で参照して、オブジェクトレベルからカメラをインポートする方法を学びます。マテリアルをすべての要素に適用し、Karma を使用してレンダリングを開始して、結果を評価します。また、キーライトとレンダリング設定を準備して、ショットの最終ルックを探求します。



**01** オブジェクトレベルに戻ります。**Tab > LOP Network** を押して、ショットをセットアップするのに使用するサブネットワークを作成します。名前を **destruction\_stage** にします。そのノードを **ダブルクリック** して、中に入ります。

デスクトップを **Solaris** に変更します。Scene View のパスバーに **obj > destruction\_stage** が表示されていることを確認します。Scene View で **D** を押して、**Background** タブで **Color Scheme** を **Dark** に設定します。

ネットワークビューで **Tab > Reference** を選択し、クリックして **reference** ノードを追加します。**Reference File** の横にある **File Chooser** をクリックし、**static\_bomb.usd** ファイルを指定します。ノードを **static\_bomb** という名前に変更します。**Primitive Path** を **/geo/\$OS** に設定します。



**02** このノードを **Alt ドラッグ** してコピーを3つ作成します。それらに **exploding\_bomb**、**ground**、**fuse** という名前を付け、**File Chooser** パラメータをこれらの USD ファイルに設定します。

**Merge** ノードを追加してこれらを接続し、**Display フラグ**を設定します。**static\_bomb** ノードの後に、**Prune LOP** を追加します。**Prune** パラメータを **\$F > 200** に設定します。**exploding\_bomb** ノードの後に、**Prune LOP** をもう1つ追加します。**Prune** パラメータを **\$F < 199** に設定します。これで、プレイヤーをスクラブすると、フレーム 200 で静的な爆弾と爆発する爆弾が切り替わります。



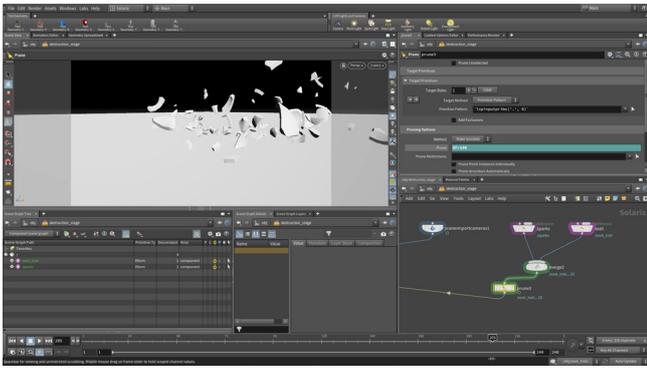
**03** **Tab > Scene Import (Cameras)** を押します。このノードを配置して、**merge** ノードに接続します。**Destination Path** を **/cam/** に設定します。これにより、オブジェクトレベルの任意のカメラを Solaris コンテキストに追加できます。

**カメラメニュー** に移動して、**cam1** を選択し、このアニメーションカメラ越しに見るようにします。merge ノードが選択されていることを確認します。プレイヤーをスクラブし、オブジェクトレベルでセットアップしたカメラのレンズを通して、導火線と爆弾がアニメートされ、爆弾が爆発することを確認します。このカメラに変更を加えた場合は、ここ Solaris で反映されます。

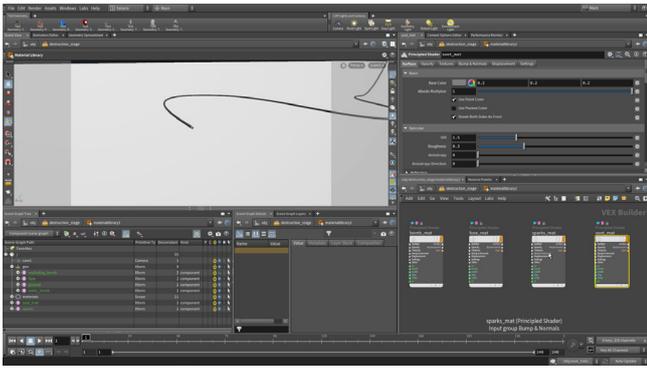


**04** ネットワークビューで **Tab > Reference** を選択し、クリックして **reference** ノードを追加します。**Reference File** の横にある **File Chooser** をクリックし、**sparks.usd** ファイルを指定します。ノードを **sparks** という名前に変更します。**Primitive Path** を **/fx/\$OS** に設定します。

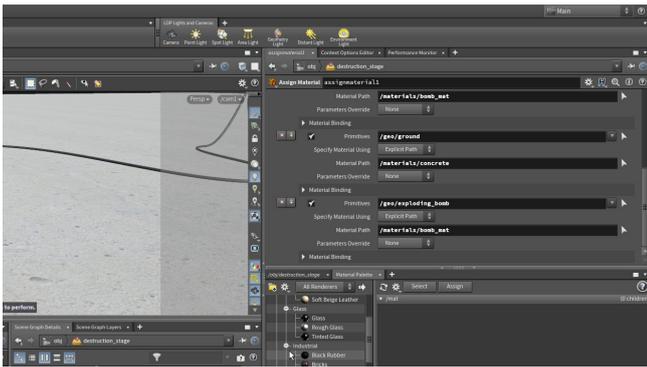
このノードを **Alt ドラッグ** してコピーを作成します。**soot** という名前を付けたら、**File Chooser** をクリックして **soot\_trail.usd** ファイルを指定します。**Merge** ノードを追加してこれらのファイルをまとめたら、それをメインの **Merge** ノードに接続し、**Display フラグ**を設定します。スクラブすると、火花が見え、火花が消えた後にすすが残るのが分かります。USD ファイルはフレーム 240 までパーティクルを維持します。



**05** エフェクト **merge** ノードの後に、**Prune LOP** を追加します。**Display フラグ**を設定します。**Prune** パラメータを **\$F >199** に設定します。これで、この時点ですと火花が消えるようになります。



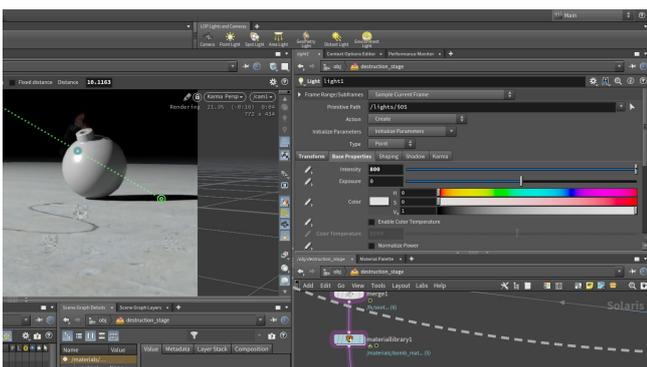
**06** **merge** ノードの後に **Material Library** ノードを追加します。**Material Palette** に移動して、**Principled Shader** と **Concrete Shader** を **/stage/materiallibrary** にドラッグします。ネットワークビューに移動して、Principled Shader の名前を **bomb\_mat** に変更します。**Alt ドラッグ** して Principled Shader のコピーを 3 つ作成し、**fuse\_mat**、**sparks\_mat**、**soot\_mat** という名前にします。



**07** 1つ上のレベルに戻り、チェーンの終端に **Assign Material** ノードを追加します。シーングラフから、**static\_bomb** と **exploding\_bomb** を **Primitives** フィールドにドラッグします。**Material Path** の横の矢印をクリックして、**bomb\_mat** を選択します。**+** (プラス) 記号を押してセクションを追加し、この手順を繰り返して **fuse**、**sparks**、**soot** にマテリアルを割り当てます。**concrete** マテリアルを **ground** に割り当てます。



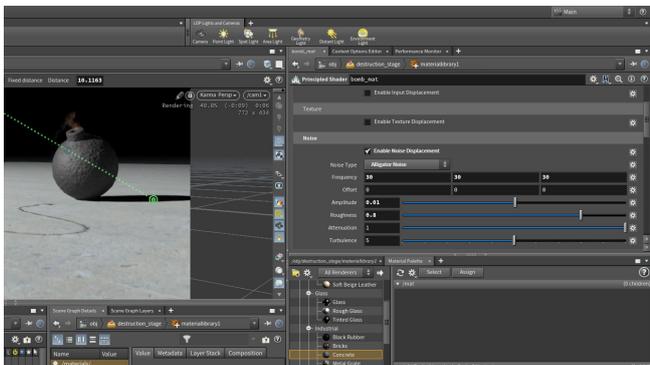
**08** フレーム 180 あたりに移動します。**LOP Lights and Cameras** シェルフの **Light** ツールをクリックします。ショットにライトが追加され、ライト越しに見えるようになります。**Base Properties** タブに移動し、**Intensity** を **50** に設定します。Scene View で、**Shadow** ボタンをクリックします。爆弾のサーフェスで **クリック** したら、その背後を **Shift クリック** してシャドウを作成します。



**09** Scene view で、**Persp** メニューから **Karma** を選択します。**Reference Plane** をオフにします。

**Material Palette** に戻り、**concrete** シェーダを選択します。**Texture** セクションで、**Effect Scale** を **0.005** に設定します。サイドバーで **Denoiser** をオンにします。グラフィックスカード (nVidia カードのみ) を使用して、レンダリング時に、ノイズが素早く解決されるようになります。



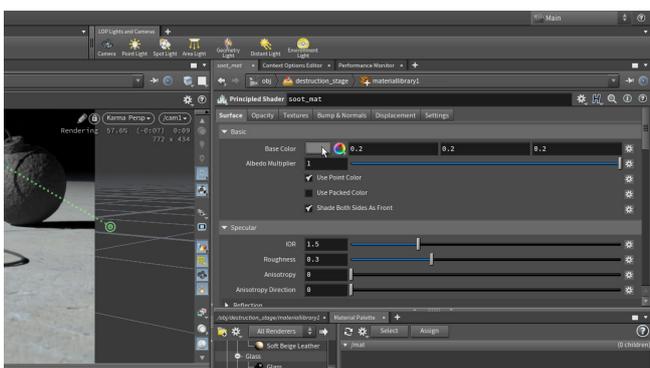


**10** *bomb\_mat* シェーダを選択します。**Surface** で次のように設定します。

- **Base Color** を黒色 (0, 0, 0) にする
- **Roughness** を 0.7 にする

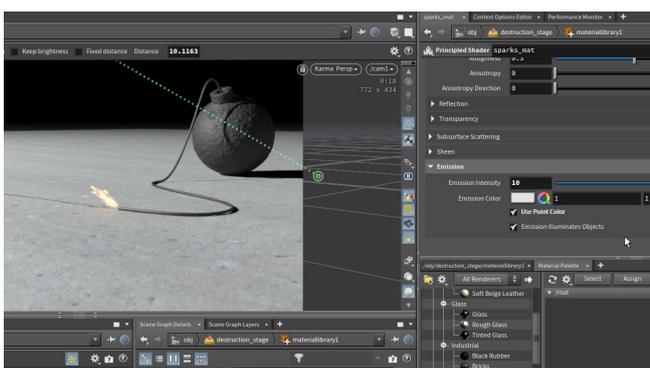
**Displacement** で、**Enable Noise Displacement** をオンにし、次のように設定します。

- **Noise Type** を **Alligator Noise** にする
- **Frequency** を 30, 30, 30 にする
- **Amplitude** を 0.01 にする
- **Roughness** を 0.8 にする



**11** *fuse\_mat* と *soot\_mat* シェーダで、**Base Color** を**ダークグレー**に設定します。

*fuse\_mat* シェーダで、**Use Point Color** をオフにして、Base Color でルックを制御できるようにします。



**12** *sparks\_mat* シェーダを選択し、**Emission** で次のように設定します。

- **Emission Color** を 1, 1, 1 (白) にする
- **Emission Intensity** を 10 にする
- **Use Point Color** をオンにする

これで火花がより明るく光り、地面も少し照明されるようになります。



**13** ステージレベルに戻ります。**ネットワークビュー**で、**Tab > Karma** を押し、**Karma Render Settings** と **USD Render ROP** ノードを追加します。それらをチェーンの終端に接続します。

*karmarendersettings* ノードを選択し、**Primary Samples** を 32 に設定します。**Image Output > Filters** タブで、**Denoiser** を **nVidia Optix Denoiser** に設定してデノイザをオンに戻します。

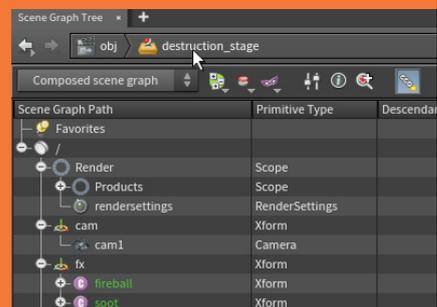
**Advanced** タブで、**Sampling** セクションに移動し、**Convergence Mode** を **Path Traced** に設定します。



## レンダリング設定

**Karma Render Settings** ノードで、シーングラフの一部となるレンダリング設定が追加されます。このノードは、ビューポートやディスクにレンダリングするのに使用されます。

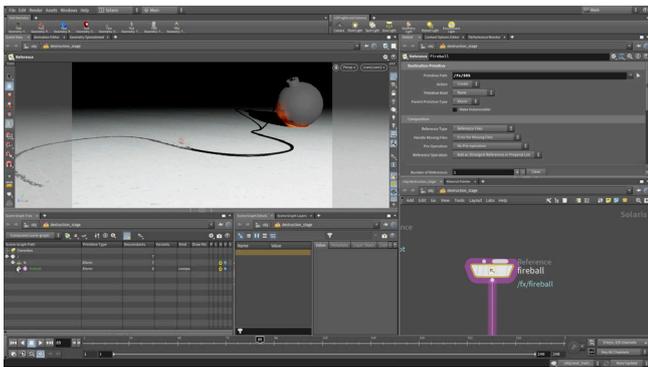
これらの設定を追加する前、ビューポートでのレンダリングにはビューポート設定が使用されていました。Scene View で Karma レンダリングをセットアップしてある場合は、**D** を押しとこれらの設定が表示されます。シーングラフにレンダリング設定がある場合は、ビューポート設定が上書きされます。



## パート 11

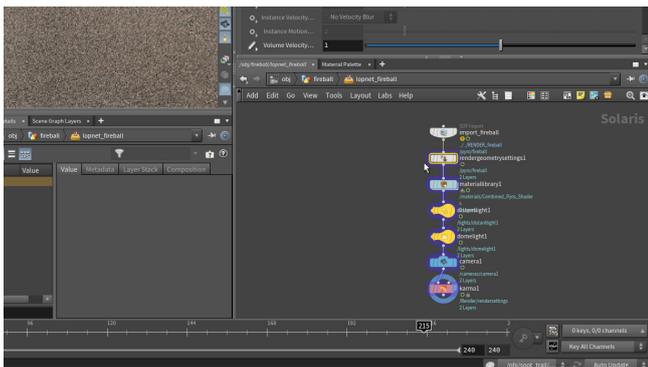
# PyroFX のレンダリング

ショットを完成させるには、火の玉の USD ファイルを追加して、適切なマテリアルを割り当てる必要があります。次に、もう 1 つカメラをセットアップして爆発の広角ショットを作成したら、2 つのシーケンスをレンダリングして最終的なシーケンスに仕上げます。その後、Mplay 画像ビューアを使用して結果をプレビューします。



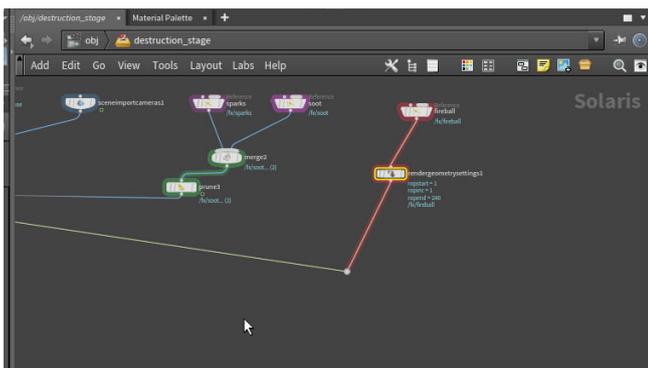
**01** Persp ビューメニューを **Houdini GL** に再度設定します。ネットワークビューで **Tab > Reference** を選択し、クリックして **Reference** ノードを追加します。**Reference File** の横にある **File Chooser** をクリックし、**pyrofx\_fireball.usd** ファイルを指定します。ノードを **fireball** という名前に変更します。**Primitive Path** を **/fx/\$OS** に設定します。

このノードを元の **merge** ノードに接続します。接続されたワイヤーを **Alt** キーを押しながらクリックしてドットを追加し、そのドットを右下に移動します。**Karma** ノードの **Display フラグ** をオンにしたまま、**フレーム 204** あたりにスクラップして爆発を確認します。



**02** オブジェクトレベルに移動して、**fireball** オブジェクトに戻ります。**pyrosolver\_fireball** ノードを選択します。**Quick Setups** メニューから、**Create Render Stage** を選択します。これにより、このネットワークに **lopnet\_fireball** が追加されます。その中に入り、推奨されるセットアップを確認します。これを単独で使用して火の玉をレンダリングすることもできますが、既存の LOP ネットワークの一部としてこのセットアップを使用する必要があります。

**rendergeometrysettings** ノードを選択し、**Ctrl + C** を押してコピーします。



**03** **destruction\_stage** LOP ネットワークに戻り、**Ctrl + V** を押してネットワークにペーストします。それを **fireball** reference ノードのすぐ下に接続します。

このノードは 2 つのことを行います。1 つ目は火の玉で Velocity モーションブラーをセットアップすることで、2 つ目はボリュームを使用してショットの照明をサポートすることです。



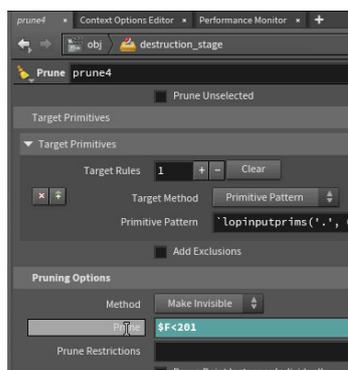
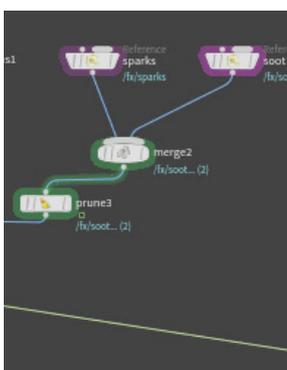
**04** **materiallibrary** ノードを選択し、**Number of Materials** の横にある + (プラス) 記号をクリックします。新しいリストの **Material VOP** の横にある **Operator Chooser** ボタンをクリックして、**fireball** オブジェクト、**lopnet\_fireball**、**matnet** と移動していき、**Pyro\_Shader** を選択します。**Accept** をクリックします。

このマテリアルは異なる LOP ネットワーク内にありますが、その位置からこの **materiallibrary** ノードに参照することができます。



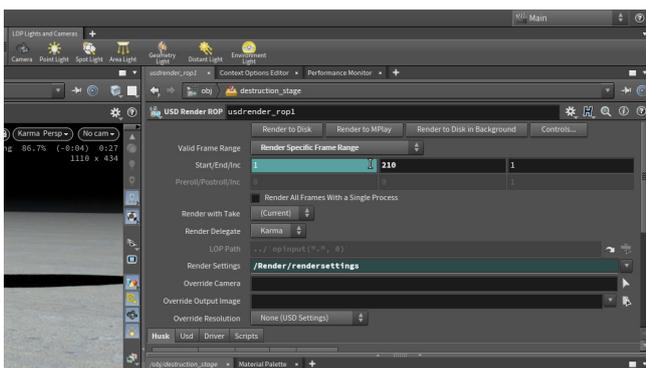
**05** `assignmaterial` ノードで、マテリアルリストをもう1つ追加します。シーングラフから `/fx/fireball` を **Primitives** セクションにドラッグして、**Material Path** の矢印をクリックし、**Pyro\_Shader** を選択します。

Persp ビューメニューを **Karma** に再度設定します。



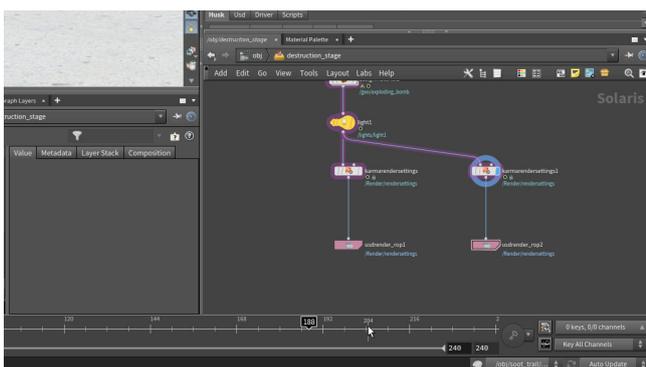
**06** ネットワークに **Prune** ノードを追加し、`rendergeometrysettings` ノードの下に接続します。Pruning Options で、**Prune** を **\$F<201** に設定します。

フレーム 200 で火の玉が爆弾のジオメトリから出ていましたが、これにより爆発が1フレーム後ろにずれます。

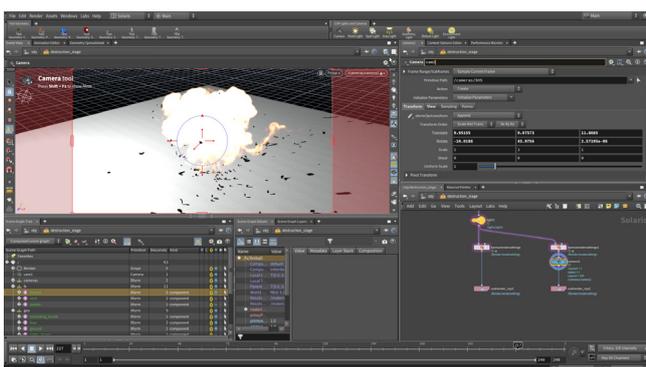


**07** Karma ノードで、**Valid Frame Range** を **Render Frame Range** に設定します。240 に設定されている **End** 値を **RMB** クリックし、**Delete Channels** を選択します。**End** 値を **210** に変更します。

最初の 210 フレームはアニメーションカメラを使用し、最後の 30 フレームは別のカメラに切り替えます。

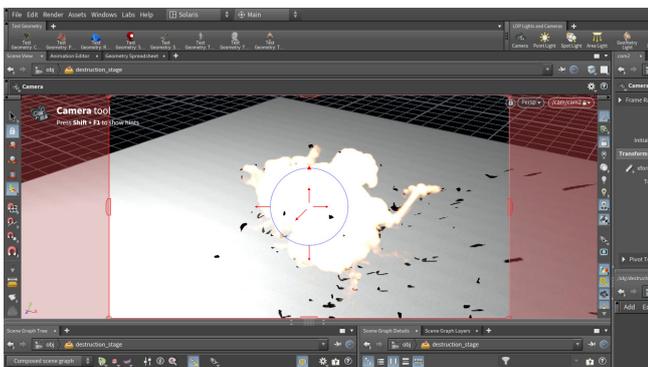


**08** Persp ビューメニューを **Houdini GL** に戻します。ネットワークビューで、`karmarendersettings` と `usdrender_rop` ノードを右側に **Alt** ドラッグします。**Display** フラグを設定します。

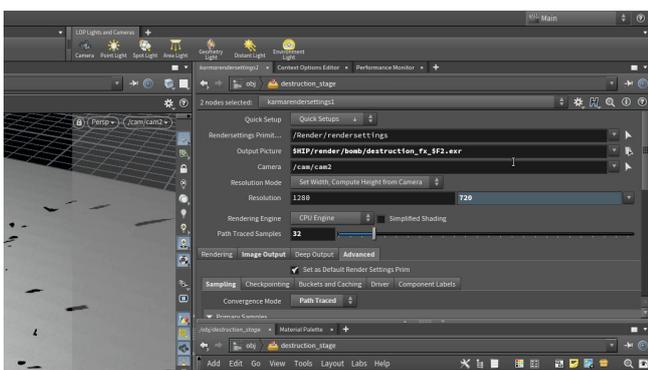


**09** Scene View で**タンブル**し、爆発を上から見下ろす新しいカメラアングルにします。LOP Lights and Cameras シェルフで **Ctrl** キーを押しながら **Camera** ツールをクリックします。

**Primitive Path** を `/cam/$OS` に設定し、名前を `cam2` にします。



**10** **Lock Camera to View** をオンにし、ビューポイントを微調整して目的のショットにします。210 と 240 の間さまざまなポイントを確認して、準備ができれば **Lock Camera to View** オプションをオフにします。



**11** **cam2** ノードを **karmarendersettings2** ノードの上に移動します。**usdrender\_rop2** ノードで、**Start** と **End** を **211** と **240** に変更し、Camera を **/cameras/cam2** にします。

1つ目の **karmarendersettings** ノードを選択し、Camera が **/cam/cam1** に設定されていることを確認します。そうでない場合、デフォルトの **camera1** はシーン内にはないので、ノードはレンダリングしません。

両方の **karmarendersettings** ノードを選択し、Output Picture を **\$HIP/render/bomb/destruction\_fx\_\$.exr** に変更します。



**12** **usdrender\_rop1** ノードを選択します。**Render to Disk** ボタンをクリックします。**usdrender\_rop2** でも繰り返します。

**Render** メニューに移動し、**MPlay > Load Desk Files** を選択します。

**render/bomb** ディレクトリに移動し、画像シーケンスを選択して、**Load** をクリックします。これにより、1つのアニメーションとして画像が再生されます。

## まとめ

パーティクル、リジッドボディダイナミクス、Pyro FX を使用して破壊ショットを作成しました。ゼロからすべてのプロジェクトを作成し、Houdini アーティストが日常的に使用するさまざまなツールやテクニックを体験しました。

また、作業内容を Solaris に取り込んで、USD を使用してシーングラフをセットアップし、Karma でレンダリングしました。

これらのスキルを活用して、皆さん自身の破壊 FX ショットを探求してみてください。

