

Ver.6.1対応

メッシュ・ボーン機能  
解説入り！



OPTPIX

# SpriteStudio® BEGINNERS 2

誰でも作れる！2Dアニメーションの基本の“き”

発行 ウェブテクノロジー  
監修 池田陽朗

# 夢を描いて、動かそう



OPTPiX SpriteStudio（以後 SpriteStudio と表記）は、Windows と Mac に対応した超汎用 2D アニメーション作成ツールです。

その機能の多彩さと使いやすさで、2D のキャラクターモーションやエフェクト、UI アニメーションといったゲーム開発現場での活用はもちろん、ショートアニメーション作品やトレーラーなどの動画作品の作成にもご利用いただいています。

クリエイターのありとあらゆる「動かしたい」に応える。それが SpriteStudio です。

## 本書の目的と役割

本書、OPTPiX SpriteStudio BEGINNERS 2 は前回配布時にご好評いただいた「OPTPiX SpriteStudio BEGINNERS」のアップデート版です。

解説するバージョンをメッシュ・ボーン機能とマスク機能を搭載した「Ver.6.1」とし、SpriteStudio のクリエイティブ作成の基本フローを解説した PART1 と、各種機能の活用方法を解説した PART2 の二部構成となっています。

また、活用事例も掲載しておりますので、どの 2D アニメーションツールを導入するかご検討の際にも、比較資料として本書がお役に立てればと考えております。

この本を入り口に、SpriteStudio を知っていただくことが目的。

そして、SpriteStudio に触れていただいた皆様の制作活動の一助となること。

それが本書の役割です。

本書で使用している各ソフトは 2018 年 1 月時点のものを使用しています。  
解説している OPTPiX SpriteStudio は Ver.6.1.0 となります。  
最新バージョンのアップデート情報はリリースノートをご確認ください。

<https://www.webtech.co.jp/help/ja/spritestudio/releasenote/>

# OPTPiX SpriteStudio BEGINNERS 2

だれでも作れる！ 2D アニメーションの基本のき

## CONTENTS



### さあ動かしてみよう！ 『SpriteStudio の基本のき』

- 1) SpriteStudio ってどんなツール？
- 2) イラストを準備しよう
- 3) SpriteStudio を起動しよう
- 4) 実際にうごかしてみよう



### こんなこともできちゃうの？ 『SpriteStudio の多彩な機能』

- 5) 参照セルでコマアニメを取り入れよう
  - 6) メッシュ・ボーン機能でやわらかい動きを表現しよう
  - 7) マスク機能を活かした演出を作ってみよう
  - 8) エフェクト機能でパーティクルエフェクトを作ってみよう
  - 9) インスタンス機能で複数のアニメを合成させよう
- Ex-1) PSD を自動でセットアップ！ PSDtoSS
- Ex-2) SsKeyBindingEditor でショートカットを設定しよう



### SpriteStudio の活用事例

池田陽朗のアニメーション作成メイキング  
～『てのひらの宇宙』はどのようにして作られたか

PART  
1

さあ動かしてみよう！  
『SpriteStudioの基本のき』





アニメーションを作って！と OPTiX SpriteStudio を渡された……。  
けれど、一体どうすればいいかわからない……このツールはどのようなものなの？  
この章では SpriteStudio とはどのようなツールなのかはもちろん  
補間アニメーションのしくみなど、SpriteStudio を使う上での基本を学んでいきます。  
この章を読んで SpriteStudio と仲良くなりましょう！

# 1 SpriteStudio ってどんなツール？

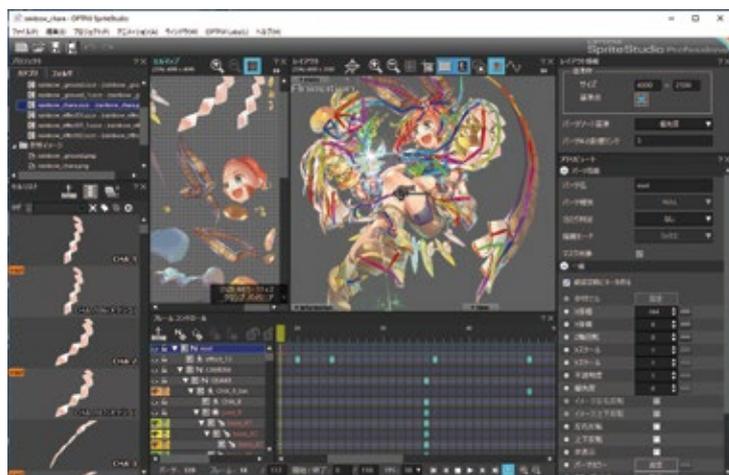
汎用2Dアニメーション作成ツール「OPTiX SpriteStudio」。そう言われてもいったいどんなことができるのか分からない……。まず「SpriteStudio」はどんなツールかをご紹介します。

## SpriteStudio はどういう場面で使われているの？

OPTiX SpriteStudio は超汎用2Dアニメーション作成ツールです。

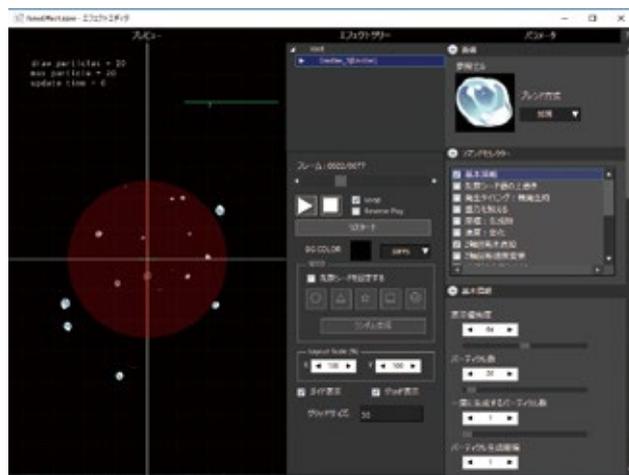
主にゲーム開発の現場で使われており、キャラクターのモーションはもちろん、魔法のエフェクトやUIの座標設定、オープニングムービー作成など、様々な用途に耐える柔軟性と機能を持ち合わせています。SpriteStudio は、作成したデータを Unity や Unreal Engine4 といったゲームエンジンで再生するためのプレイヤーや SDK を無償で公開しています。

また、インディーライセンスも発行しており、アマチュア作家の活動も支援しています。ショートアニメ作品を作ったり、インディーゲームでリッチな演出を作ったりしたり……。SpriteStudio はクリエイターの「動かしたい」をサポートし、実現することができるツールです。



シンプルな操作系で2Dキャラクターのモーションを簡単に作成。

メッシュ・ボーン機能で表現力豊かなアニメーションをうみだすことができます。



パーティクルエフェクトも「エフェクトエディタ」で簡単に作れます

SpriteStudio で作成されたアプリやゲームは日々増え続けています。

採用実績は右の QR コードからご覧いただけます。

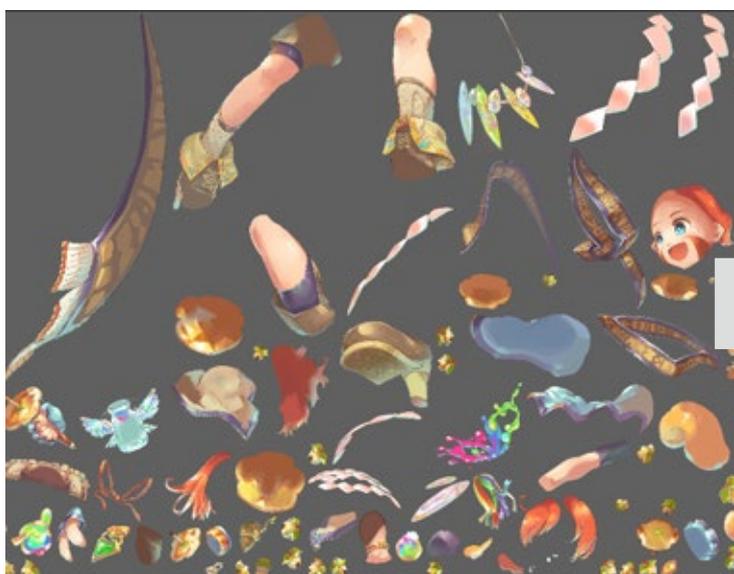
<http://www.webtech.co.jp/spritestudio/gallery/>



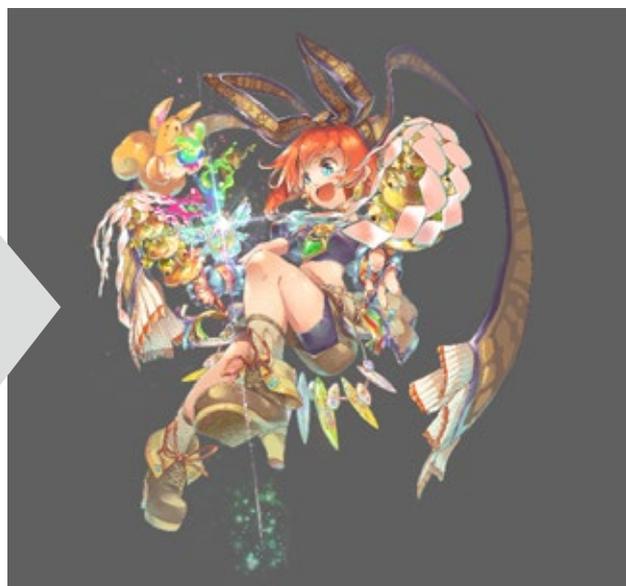
# 補間アニメーションって何？ 「SpriteStudio」のアニメーションの基本

OPTiX SpriteStudio は、分割・レイアウトしたパーツを「補間アニメーション」でアニメーションを作るツールです。補間アニメーションとは、キーを設定した間の数値を自動で計算し、アニメーションを作成する方式です。

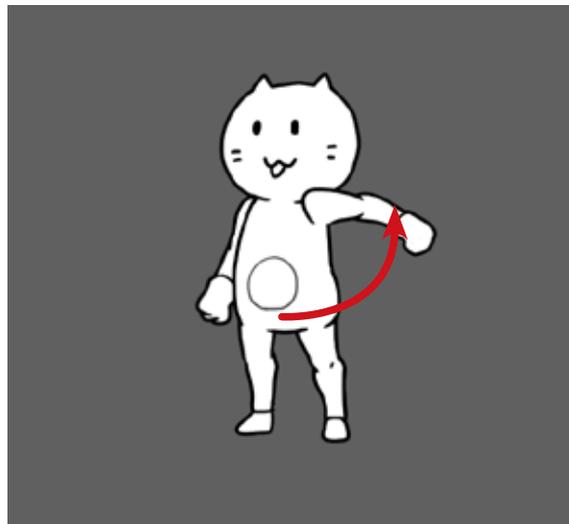
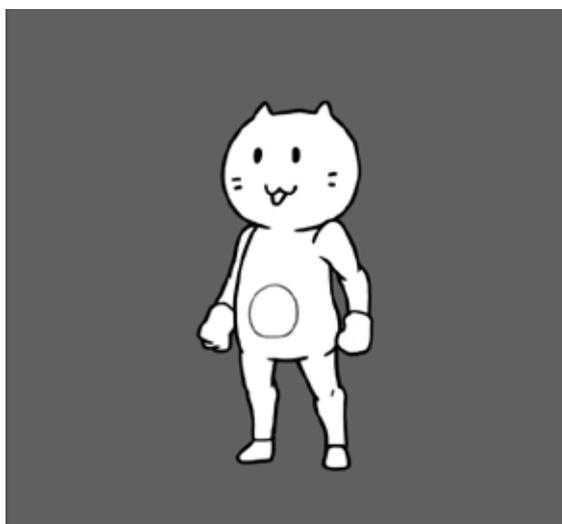
一般的なテレビアニメなどのアニメーションは「コマアニメ」と呼ばれており、中割りりりとなる絵をすべて描きだす必要がありますが、補間アニメーションは中割りりりとなる部分はすべてコンピューターが計算して作成するので、コマアニメと比較して手間がかかりません。



このように分割したパーツを……



レイアウトして、パーツをそれぞれ動かします。



AのキャラクターのポーズがBに変化する途中の動き（中割りにあたる部分）はSpriteStudioが自動で計算して動かします。そのため、中割りにあたる絵を描く必要はなく、簡単にアニメーションが作れます。

## 2 グラフィックソフトでイラストを準備しよう

OPTiX SpriteStudio は「補間アニメーション」を作成するツールです。そのため、アニメーションさせるには、イラストを「パーツ分割」し「テクスチャ」を作らなくてはなりません。SpriteStudio で動かすために必要な「素材の作り方」を解説します。

### 動かすためのイラストを準備しよう

何はともあれ、SpriteStudio で動かすための絵が必要です。

例えば、ボールの絵を回転させたり、上下に伸び縮みさせるといった簡単な動きであれば、イラストそのままを1つのパーツとして動かすことができます。

ですが、多関節の生き物や人間のキャラクターなど、部位ごとに動きを設定する必要がある場合は『パーツ分割』をしなくてはなりません。

### パーツ分割をしよう

SpriteStudio は「補間アニメーション」を作成するツールです。そのため、中割りとなる絵は不要ですが、動かしたいところを「パーツ」に分割する作業が必要です。

下図のキャラクターの場合、肩、腕、手、首といった関節部分と髪やリボンといった揺れるものを主に分けています。

特に顔や髪の毛はよく注目されるので、多めに分割しています。



## 空白部分を描き足そう

イラストがあらかじめ「パーツ分割されていない一枚絵」の場合、分割した際にできる空白部分を書き足す必要があります。

例えば「腕を振る」動きを付けたい場合、腕パーツを切り取るとその部分のマントが空白になります。

この状態で腕を振り上げると空白部分が見えてしまうため、不足している部分は描き足す必要があります。



この作業を繰り返し、パーツを分割していきます。

パーツはアニメーションを作りながら分割しなおすことが可能なので、あまり悩まずに「動かしたいところ」を想像しながら分割してみましょう。

# 取り込むための「テクスチャ」を作ろう

分割したパーツを一枚の画像に配置して「テクスチャ」を作りましょう。

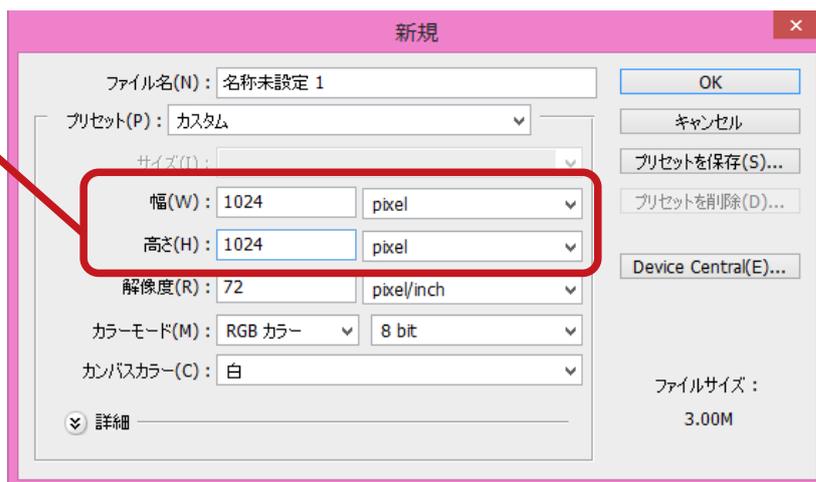
この作業もグラフィックソフトで行います。

ファイルを新規作成します。この際、サイズは2のべき乗にする必要があります。

SpriteStudio 上では2のべき乗ではないサイズの画像も一部機能を除いて編集することができますが  
ハードウェア・ゲームエンジン上では

2のべき乗ではないサイズのデータを受け付けない場合があります。

2のべき乗である1024×1024で  
ファイルを新規作成します



## HINT 2のべき乗とは？

2のべき乗はコンピューターにとって都合のよいサイズです。

ゲームエンジンによっては、2のべき乗以外で作成されたテクスチャは、うまく読み込めないなどの問題が発生することがあります。

SpriteStudio でも、エフェクト機能やメッシュ・ボーン機能では、2のべき乗以外のテクスチャを使用すると画像を表示できません。

2のべき乗は「64・128・256・512・1024・2048・4096」です。

2のべき乗であればよいため、正方形にする必要はありません。

例えば、1024×2048のテクスチャでも問題ありません。

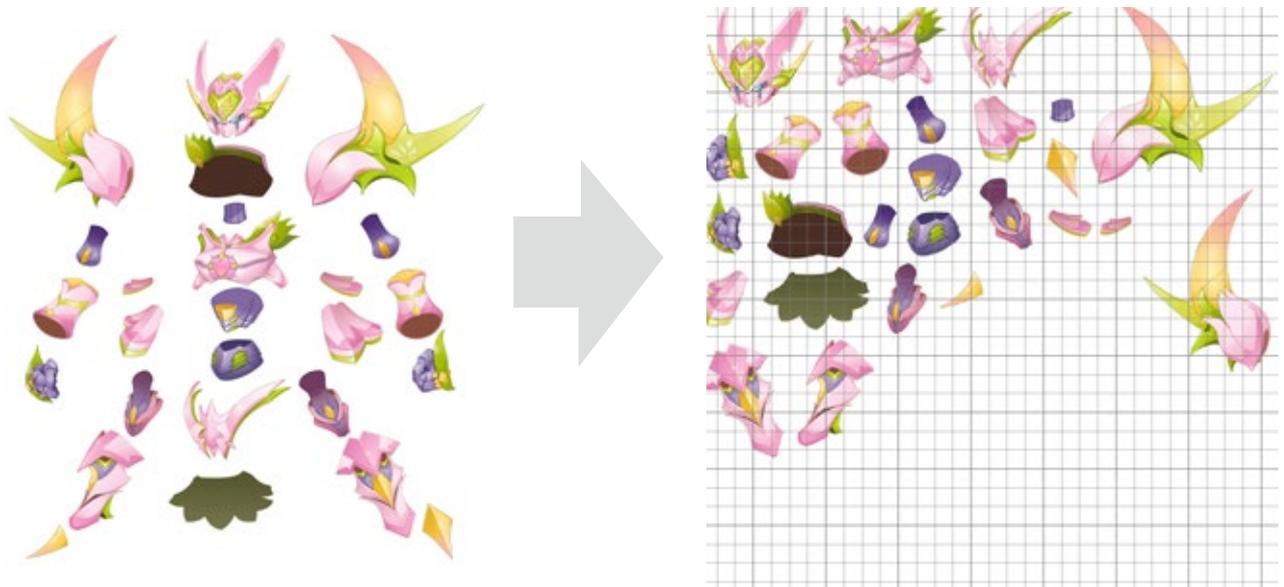
テクスチャの大きさやパーツ配置について  
詳しくはヘルプセンターもご覧ください。

<http://www.webtech.co.jp/help/ja/spritestudio/guide/tips/texture/>

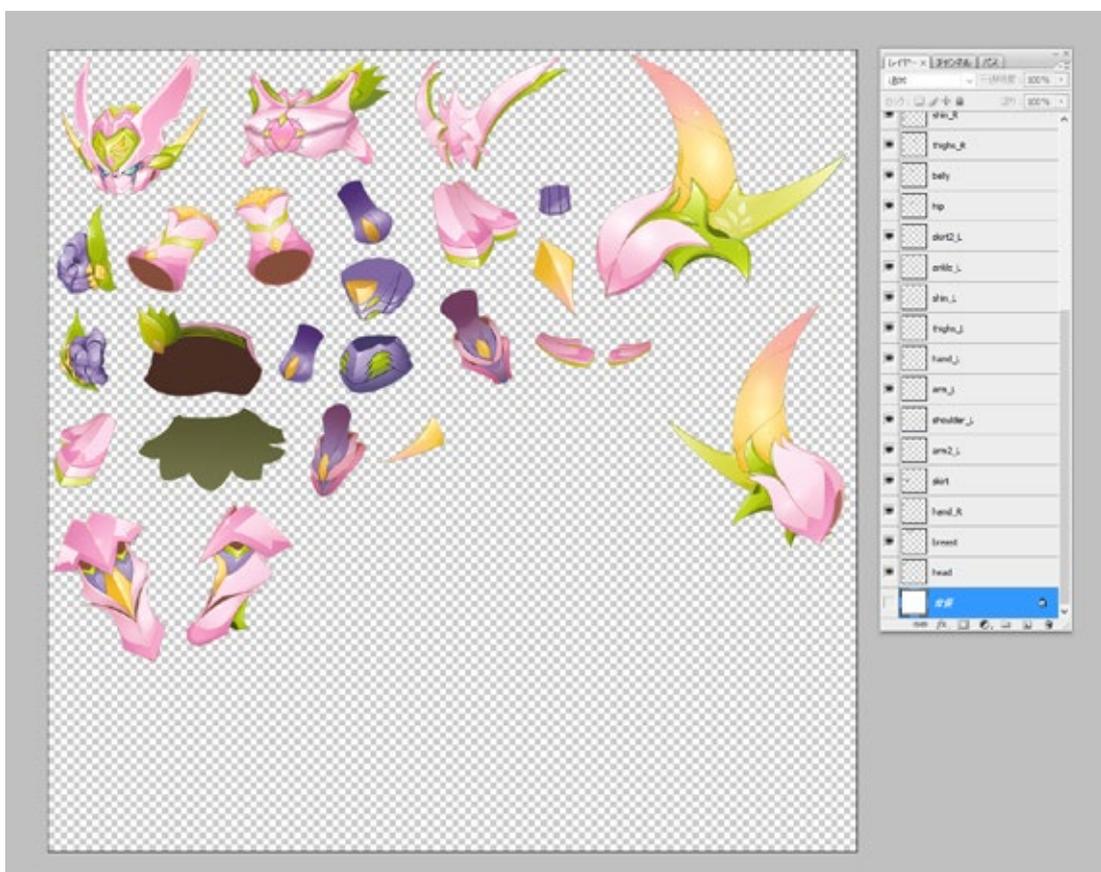


新規作成したファイルへ、分割したパーツを配置していきます。

パーツを配置する際、グリッドを表示して矩形に収まるように配置すると、SpriteStudio でセルを切り出すときに、作業がスムーズになります。



背景を透明状態にして保存します。これでテクスチャは完成です。



# 3 SpriteStudio を起動しよう

テクスチャを用意したら、SpriteStudio を起動しましょう。  
SpriteStudio のファイル形式やウィンドウの構成と機能を、前の項目で作成したテクスチャの読み込み方やレイアウトの仕方を解説しながら学んでいきます。

## 1 メインメニュー

SpriteStudio 全体に対しての  
操作を行う機能が用意されています

## 2 プロジェクトウィンドウ

作業中のアニメーションの構成を確認・  
選択するためのウィンドウです

## 3 セルマップウィンドウ

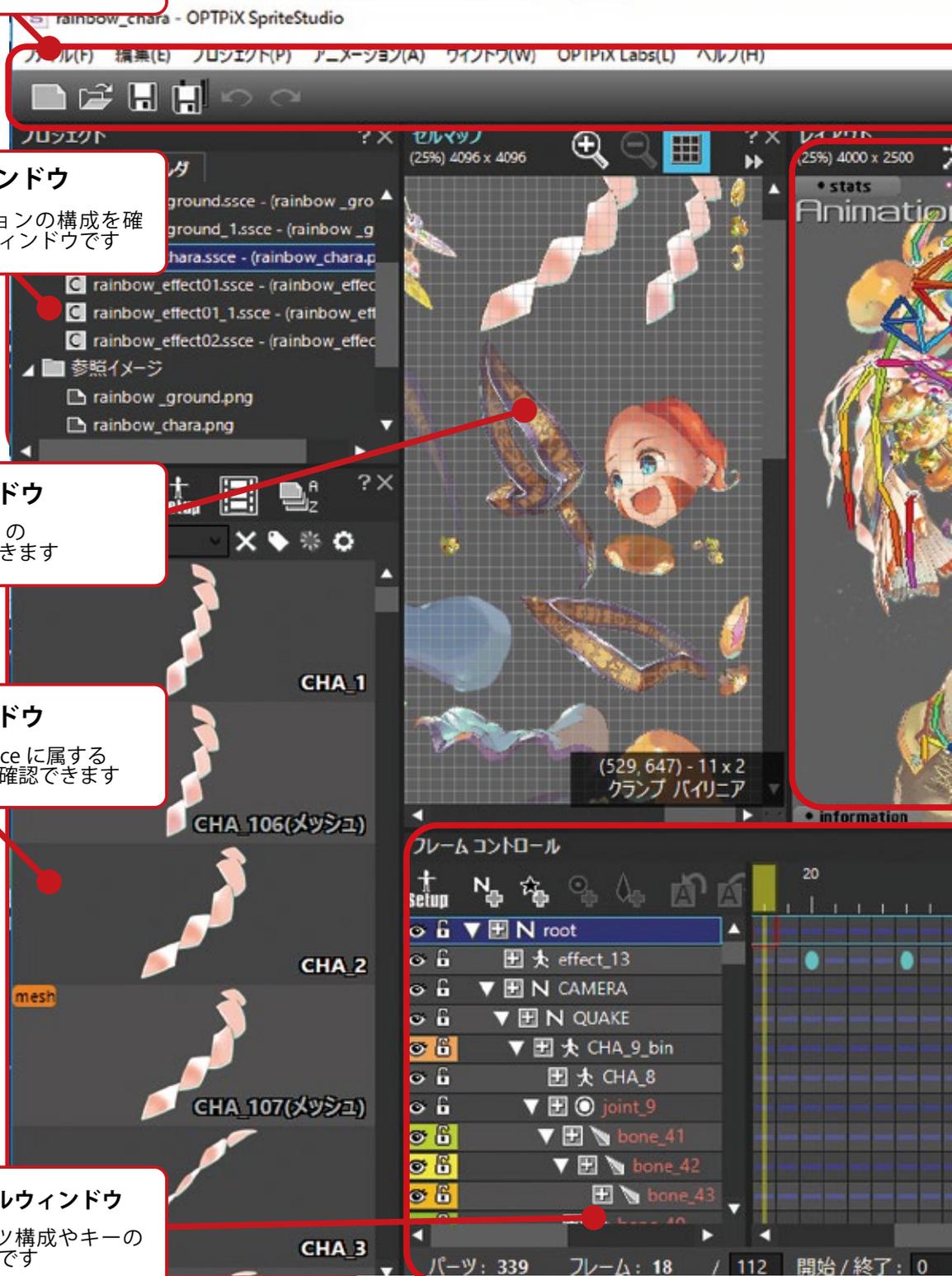
現在選択している ssce の  
参照イメージを確認できます

## 4 セルリストウィンドウ

現在選択されている ssce に属する  
セル・アニメの一覧を確認できます

## 5 フレームコントロールウィンドウ

作業中の ssae のパーツ構成やキーの  
編集を行うウィンドウです



# SpriteStudio のウィンドウについて

テクスチャを用意したら、SpriteStudio を起動してみましょう。

このような画面が表示されます。このウィンドウ全体を「メインウィンドウ」と呼びます。

「メインウィンドウ」は様々な機能のウィンドウで構成されています。

## 6 レイアウトウィンドウ

パーツの配置、アニメーションのプレビューを確認できます

## 7 レイアウト情報ウィンドウ

アニメーションのサイズや基準点など、レイアウトに関する設定を行うことができます

## 8 アトリビュートウィンドウ

選択しているキーのパラメータを設定・確認することができます



# ファイルを作ってみよう

まずはファイルを作ってみましょう。

SpriteStudio は主に3つのファイルで構成されています。

## sspj( プロジェクトファイル)

アニメーションの基準となるファイルです。

このファイルを開いて作業を開始します。

プロジェクト設定といった全体にかかわる設定情報と、  
どのようなファイルが紐づいているかなどを含んでおり、  
このファイルが無ければ作業は始まりません。



## ssae (アニメーションファイル)

レイアウトやアニメーションの情報を保存するファイルです。

複数の「アニメーションデータ」を保有することができます。



## ssce (セルマップファイル)

セルマップ、セルの情報を保存するファイルです。

セルマップは「テクスチャ」、セルはそのテクスチャを  
パーツごとに分割したデータを指します。

主に動かすための『画像の情報』を保持するファイルです。



エフェクト機能を使う場合は上記のほかに「ssee (エフェクトファイル)」が追加されます。

## ssee (エフェクトファイル)

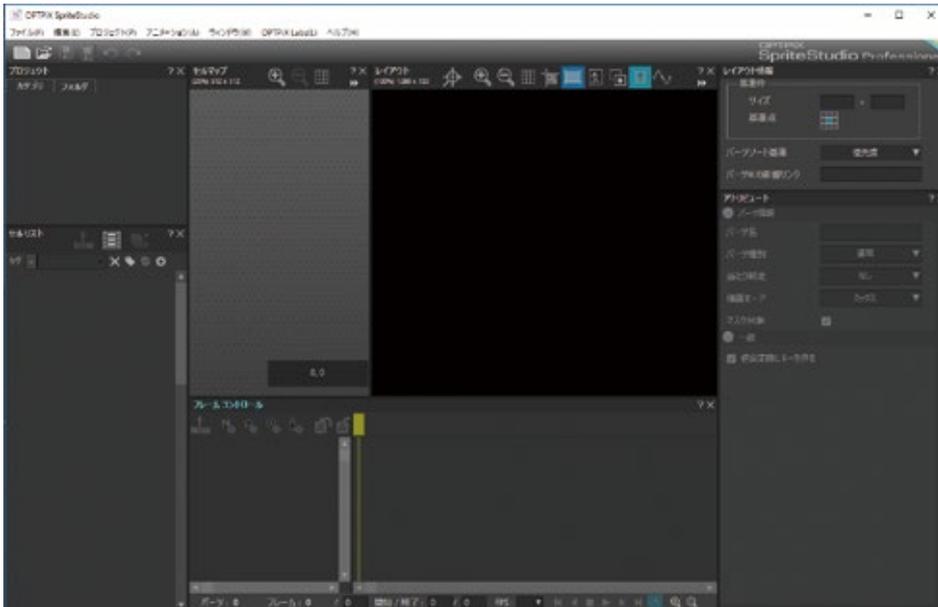
エフェクト機能を使用する際に作成するファイルです。

エフェクトの情報を保持しています。

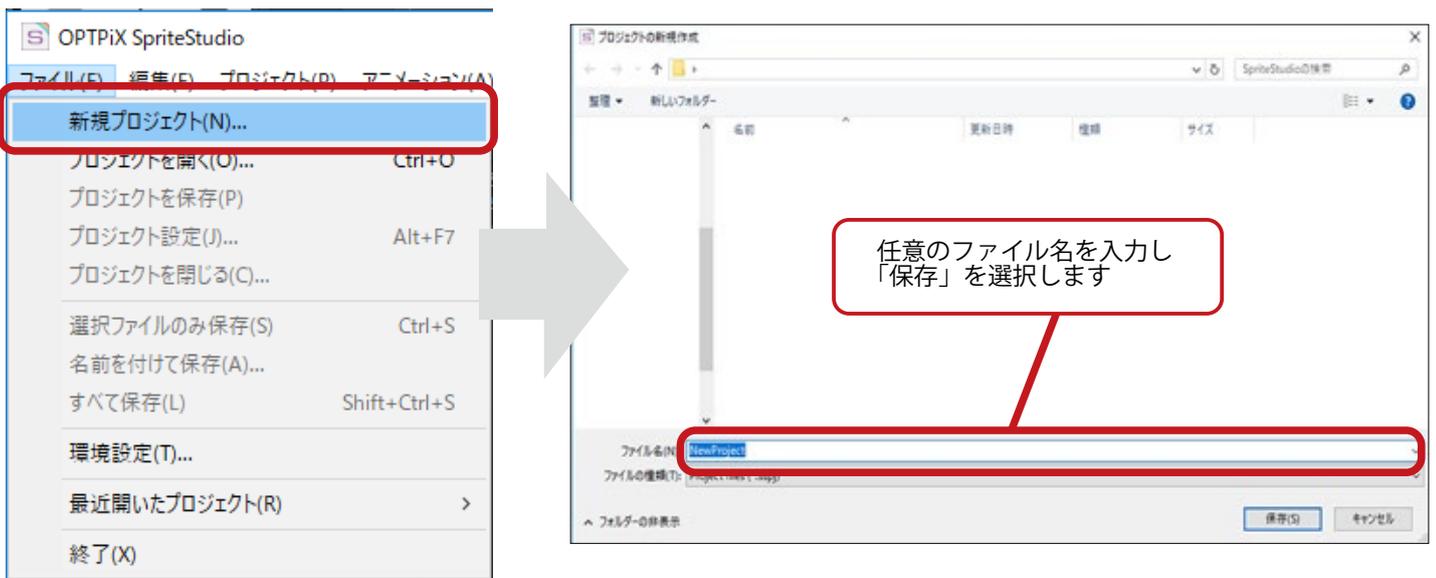
このファイルについては「エフェクト機能の使い方」の項で  
詳しく解説します。



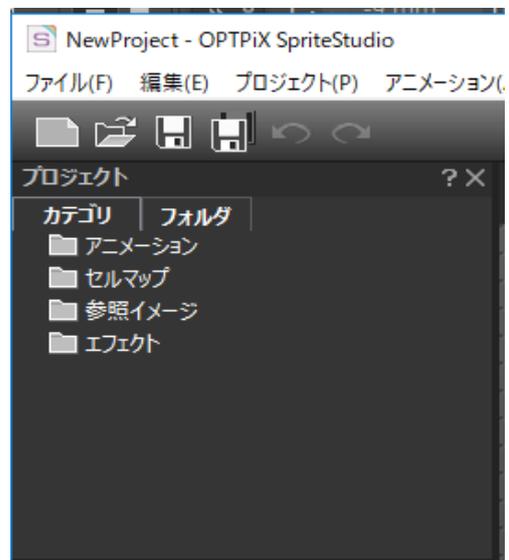
SpriteStudio を起動するとこのようなウィンドウが表示されます。



まず sspj を作成します。このファイルの中に「ssae」や「ssce」を作成していきます。  
メインメニューの「ファイル」から「新規プロジェクト」を選択し、任意のファイル名を設定します。

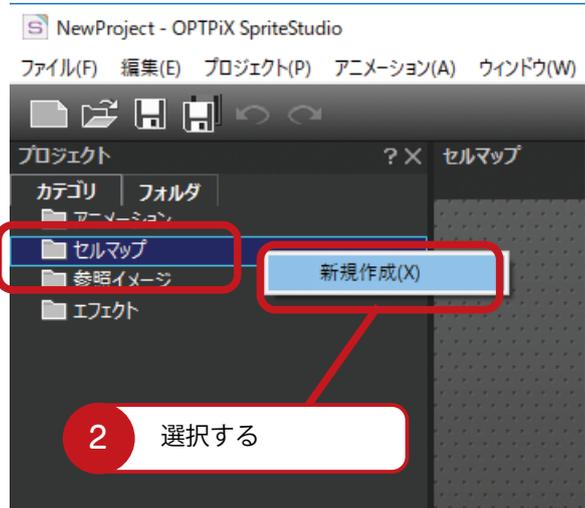


sspj が完成しました。



次に「ssce」を作成します。「プロジェクトウィンドウ」の「セルマップ」を右クリックし、新規作成を選択します。

1 右クリックする



2 選択する

テクスチャを選択し、「開く」を選択します。

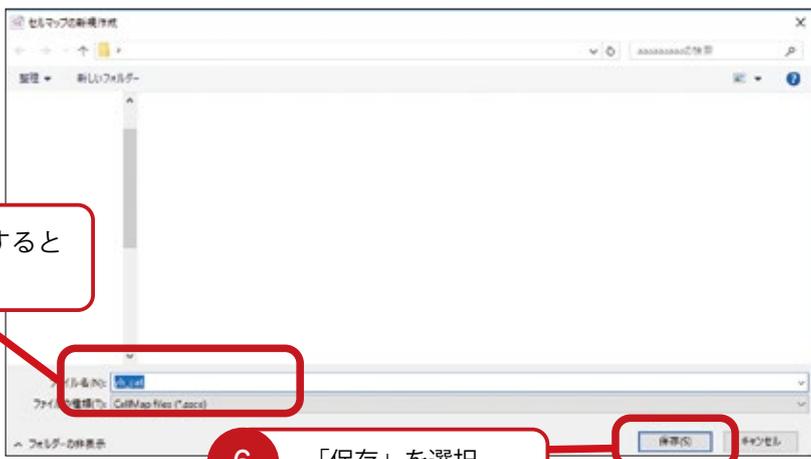
3 選択する



4 「開く」を選択

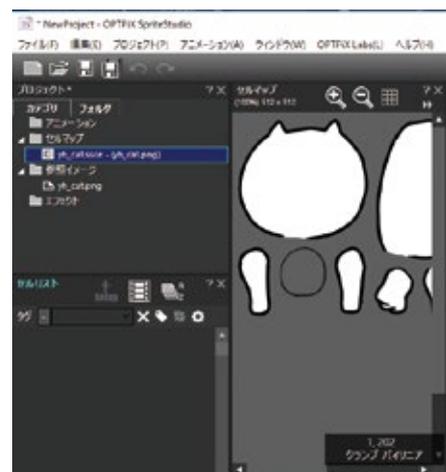
任意のファイル名を設定します。

5 任意のファイル名を入力すると ssce の名前になります



6 「保存」を選択

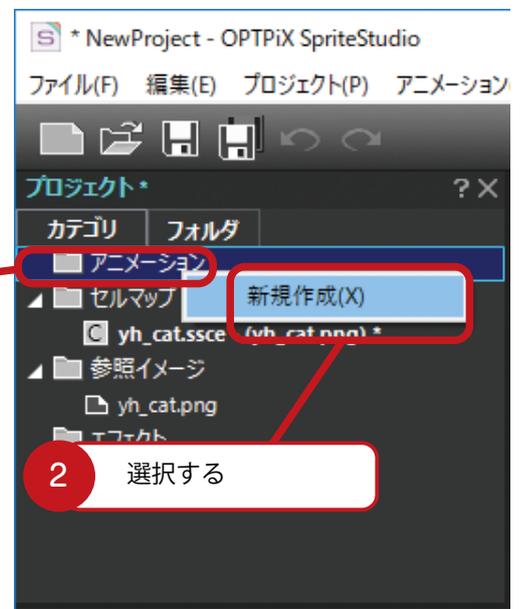
保存すると ssce が出来上がり、セルマップウィンドウに選択した画像が表示されます。



次に ssae を作成します。

「プロジェクトウィンドウ」の「アニメーション」を右クリックし、新規作成を選択します。

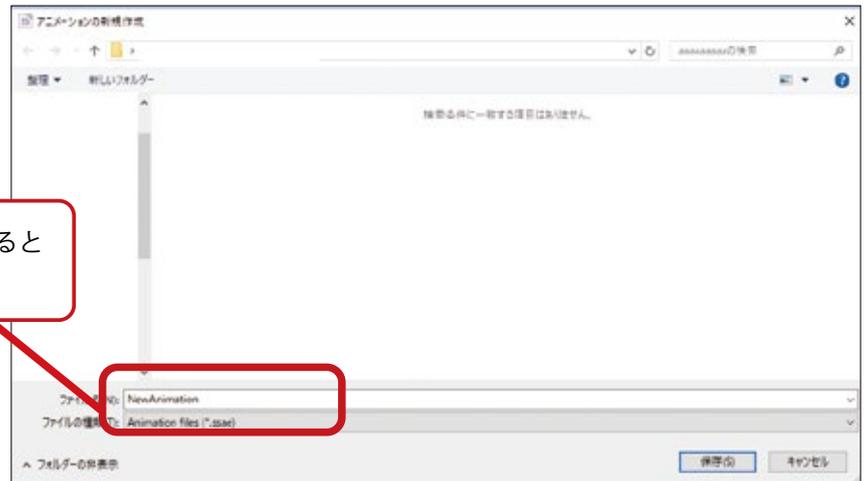
1 右クリックする



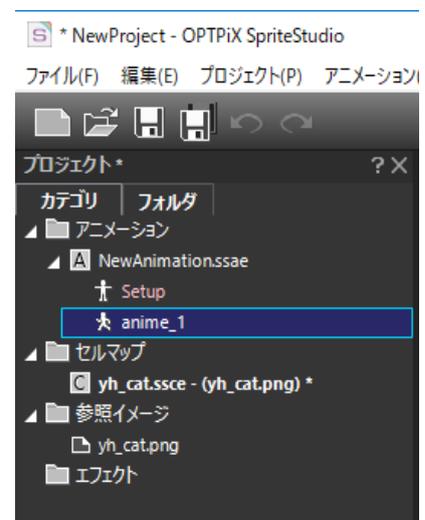
2 選択する

任意のファイル名を設定します。

3 任意のファイル名を入力すると ssae の名前になります



保存すると ssae が出来上がり  
レイアウトウィンドウ  
フレームコントロールウィンドウが表示されます。



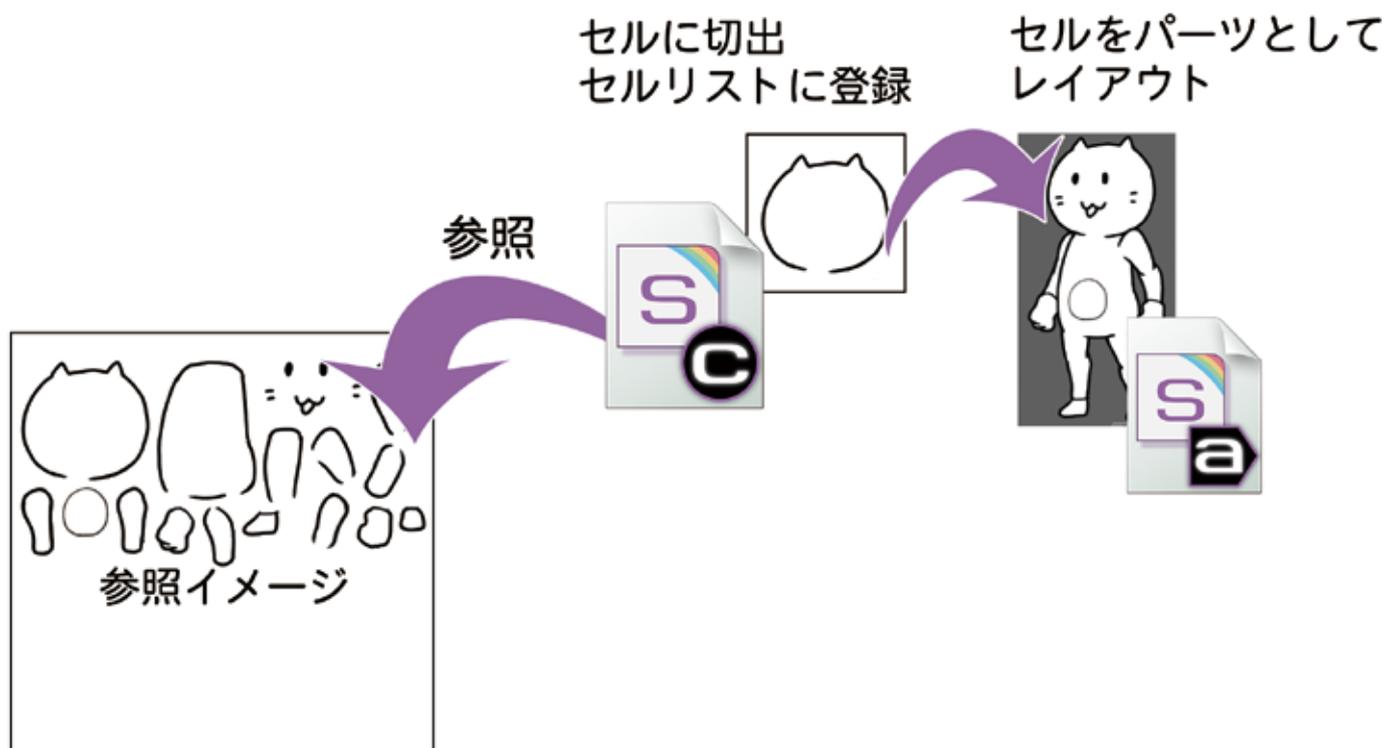
これでファイルの作成は完了です。

## 「セルマップ」から「セル」を登録しよう

SpriteStudio では読み込んだテクスチャを「参照イメージ」と呼びます。

「セルマップ」とは参照イメージを登録した「セル」のおおもととなるファイルです。

セルマップからパーツになる「セル」を作成し、セルを「パーツ」としてレイアウトしていきます。



## HINT なぜセルを登録するのか

セルマップから直接パーツとしてレイアウトしたほうが早いのに……。

手順を見てそう思った方もいると思います。

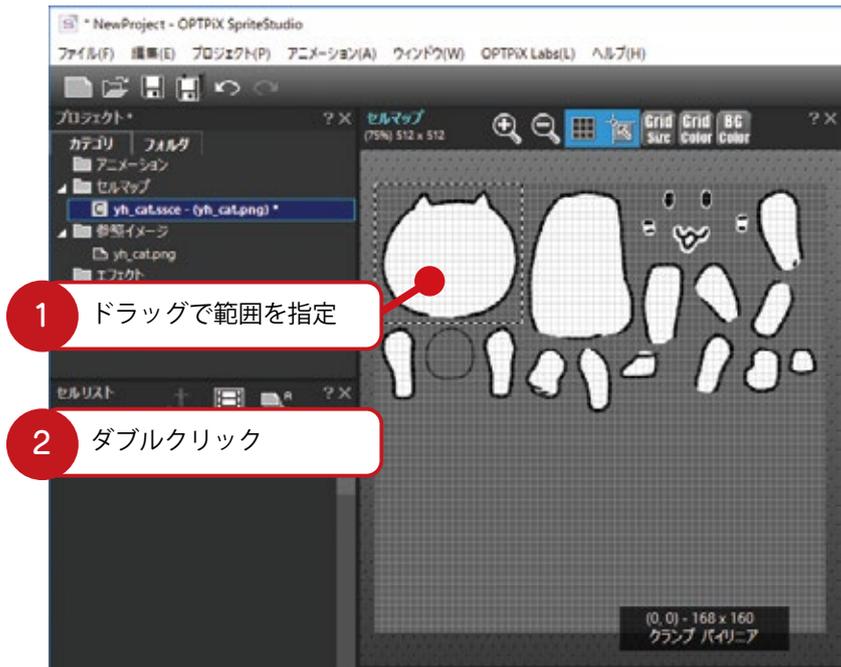
コマアニメを作成したいときにはセルを切り替えることで簡単にパーツの図柄を変更できます。

パーティクルエフェクトの図柄もセルを参照します。

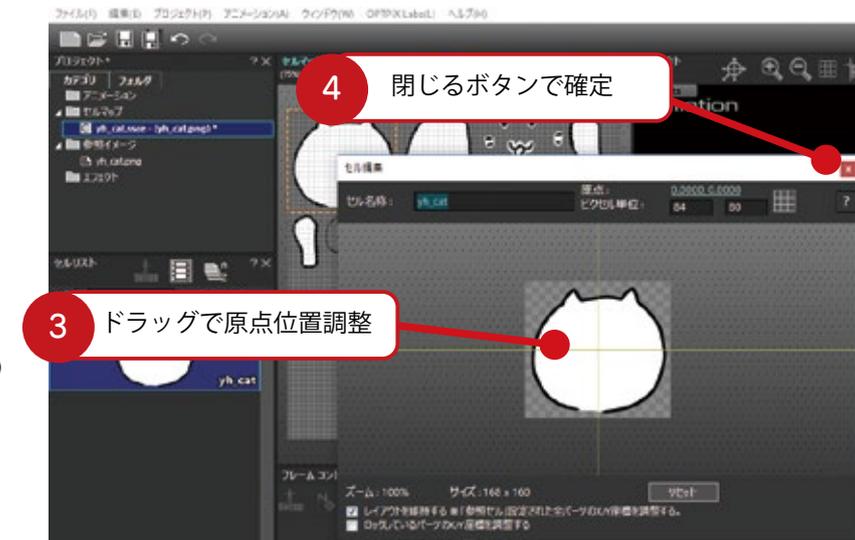
セルにしてアニメーションを組むことには様々なメリットがあります。

セルマップウィンドウで「セル」単位に切り取っていきます。グリッド数をグラフィックソフトで指定した数値に設定し、グリッドにスナップを有効にすると作業がスムーズになります。

マウスでクリック→ドラッグで切り取る範囲を設定します。範囲を決めたらダブルクリックをします。

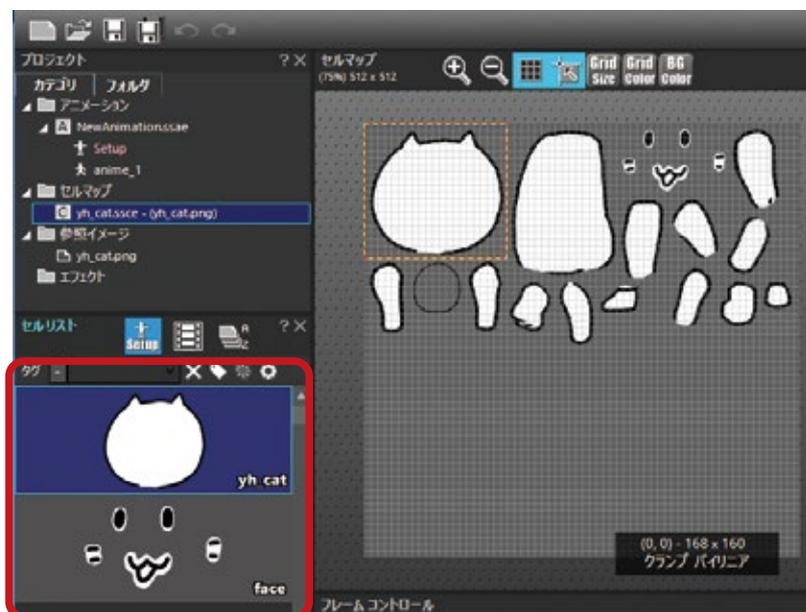


切り取り範囲が確定し、セル編集ウィンドウが開きます。セル編集ウィンドウではセルの原点を変更することができます。



セルの原点はアニメーションするときの基準となる位置です。原点を基準にパーツは動きます。

セルリストにセルが登録されました。



# HINT セットアップデータとアニメーションデータ

SpriteStudio Ver.6 から「セットアップデータ」と「アニメーションデータ」が追加されました。

「セットアップデータ」はアニメーションデータのベースとなる情報を有するデータです。  
セットアップデータで構築したパーツ構造をもとに、「アニメーションデータ」が作成されます。

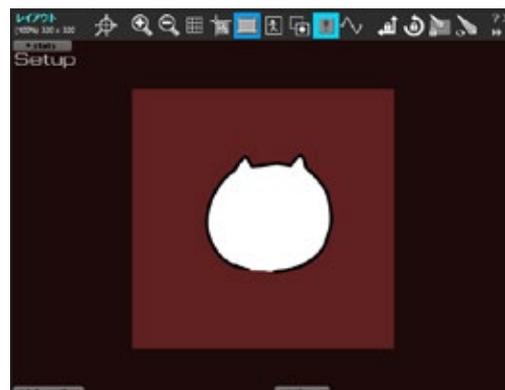
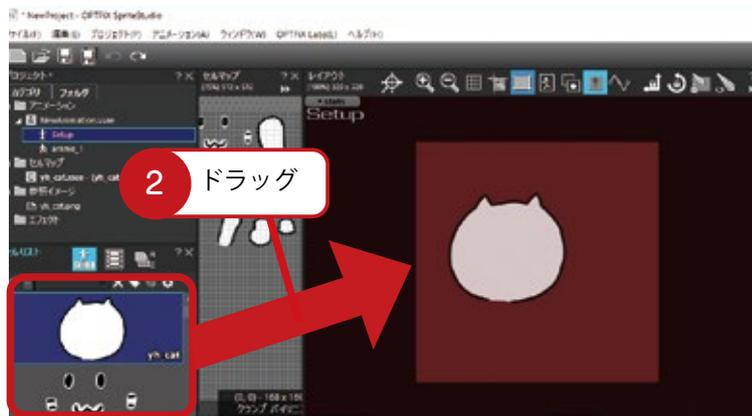
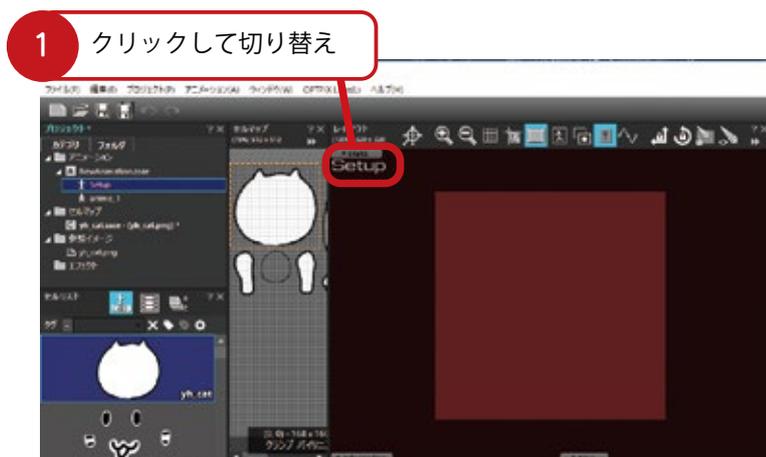
「アニメーションデータ」は、実際にアニメーションを作成するデータです。  
ssae 内に複数作成することが可能で、セットアップデータをもとに作成されます。

セットアップデータに切り替えてください。  
セットアップデータへの切り替えは、  
プロジェクトウィンドウで「Setup」を  
選択するか、レイアウトウィンドウの左上隅の  
文字をクリックすることで  
切り替えることができます。

レイアウトウィンドウの背景が赤い状態が  
「セットアップデータ」を選択している状態です。

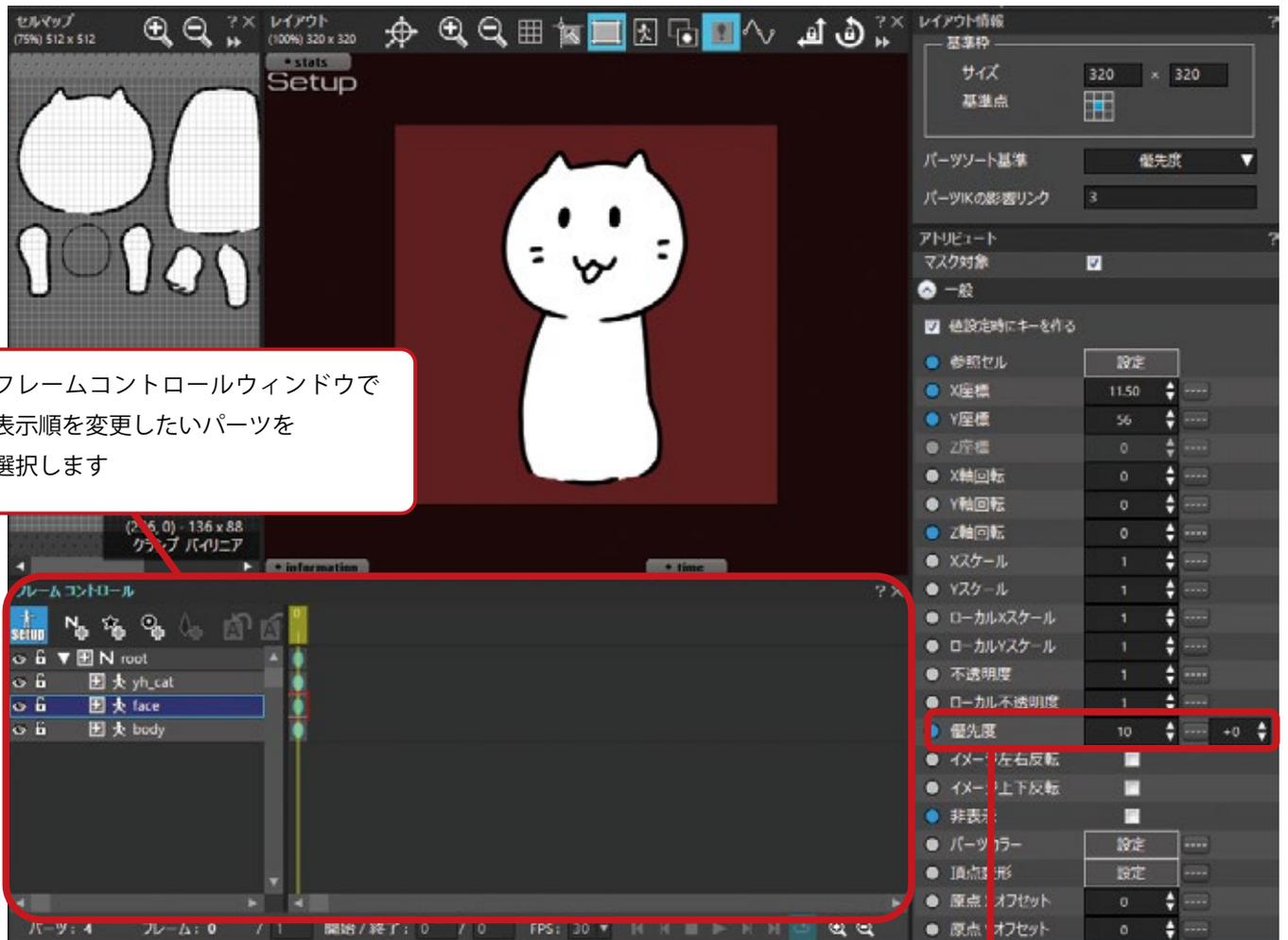
セルリストウィンドウから  
レイアウトしたいセルを  
「レイアウトウィンドウ」に  
ドラッグしていきます。  
するとセルがパーツとして配置されます。

レイアウトウィンドウに  
セルがパーツとして配置されました。  
この作業を繰り返し、  
すべてのパーツをレイアウトしてください。



パーツの表示順を設定します。  
パーツの表示順は「優先度」によって決まります。

「優先度」はアトリビュートウィンドウで設定することができます。  
「優先度」の数値が大きいほどパーツは手前に、数値が少ないほど奥に表示されます。



1 フレームコントロールウィンドウで表示順を変更したいパーツを選択します

2 「優先度」を変更します

全てのパーツの「優先度」を設定し終わったらレイアウトの完了です。  
アニメーションさせる準備が整いました。

## 4 アニメーションさせてみよう

レイアウトまで完成したら、いよいよアニメーションを作る作業のスタートです。

SpriteStudio は多彩な機能で様々な表現を作ることができますが、まずは基本から学んでいきましょう。

### まずはプロジェクト設定を確認しよう

さあ早速アニメーションを……と行きたいところですが、その前に「プロジェクト設定」を確認しましょう。

これは用途によって使用できるアトリビュートが異なるためです。

本格的にアニメーションを作る前に必ず確認してください。

特にゲーム開発の場合は、使用するゲームエンジンや仕様によって互換性が無い

(使用できない・再現できない) アトリビュートがあったり、

仕様であらかじめフレームレートが決まっている場合が多いです。

作った後に「互換性の無いアトリビュートを使用していた！」という事態に陥ってしまうと、

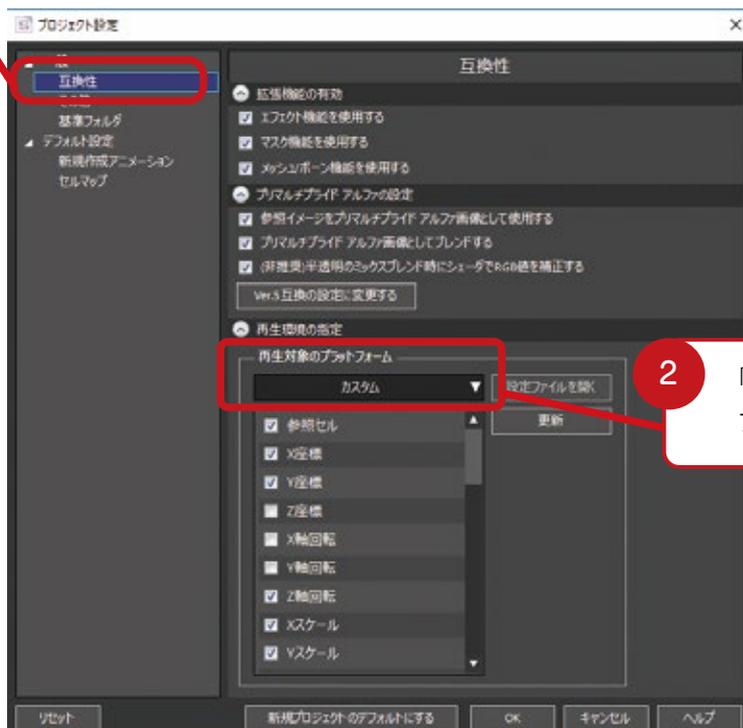
修正が非常に大変です。

**必ずプロジェクト設定を確認してからアニメーションを作りましょう。**

### 「互換性」を確認してアトリビュートを整理しよう

メインメニューのファイルから「プロジェクト設定ウィンドウ」を開きます。

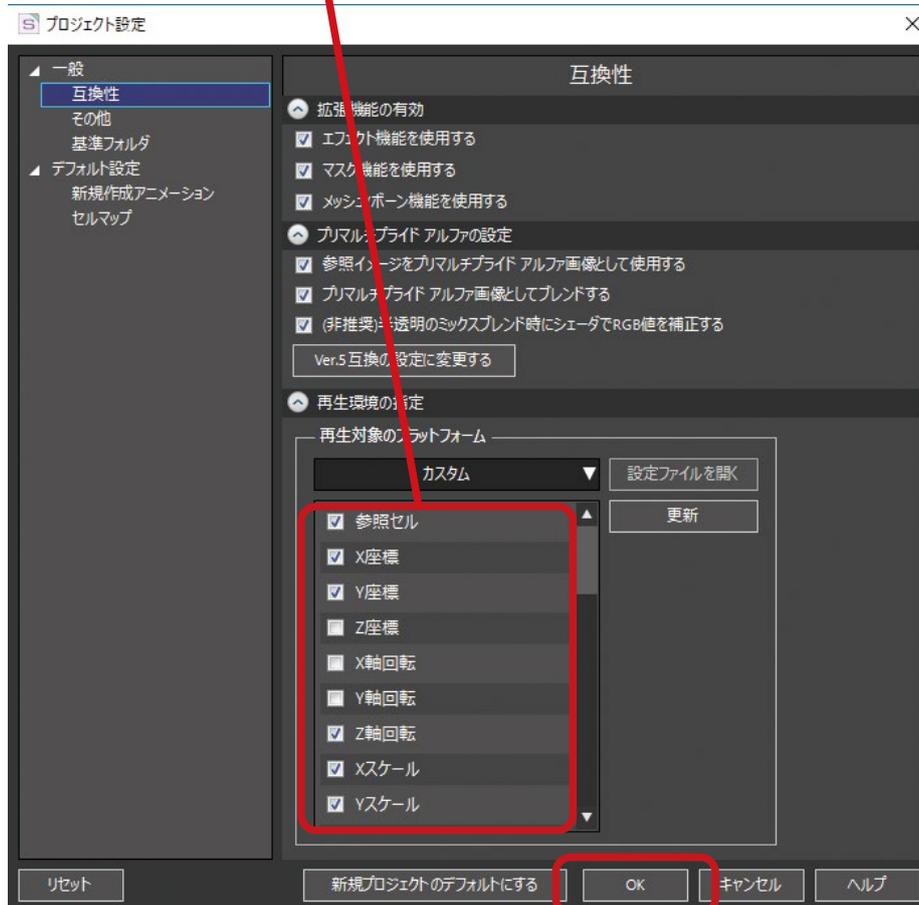
1 「互換性」を選択します



2 「再生対象のプラットフォーム」を確認します

3

今回は自由に表示するアトリビュートを選択したいので「カスタム」を選択しました。  
用途に応じてアトリビュートのチェックを外します



4

「OK」をクリックして  
設定を完了します

## フレームを確認しよう

fps（frames per second =フレーム毎秒）を確認します。

fps とは 1 秒あたりにどのくらいのフレーム数を処理するかの単位で、

24fps の場合、1 秒につき 24 フレームを処理します。

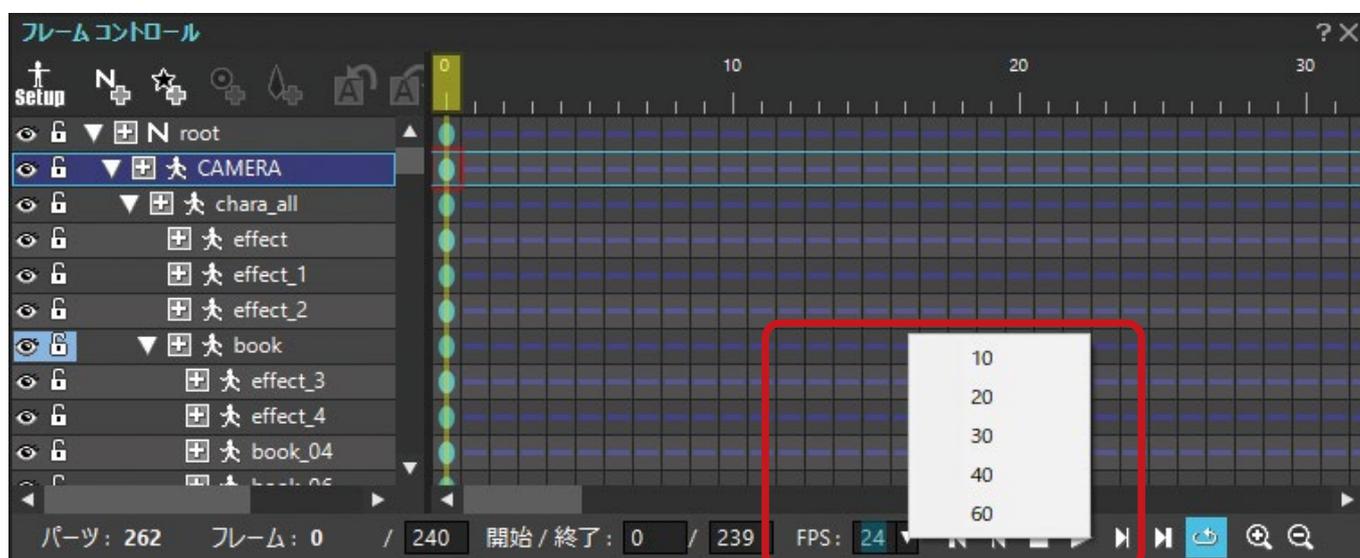
一般的に fps を大きくすればするほど、アニメーションは滑らかになりますが処理は重くなり、

小さくするほどアニメーションはカクカクしますが処理は軽くなります。

仕様があらかじめ決まっている場合、fps を仕様に合わせて設定してください。

フレームコントロールウィンドウの「FPS」の欄に任意の数値を入力するか、

プルダウンボックスから数値を選択してください。

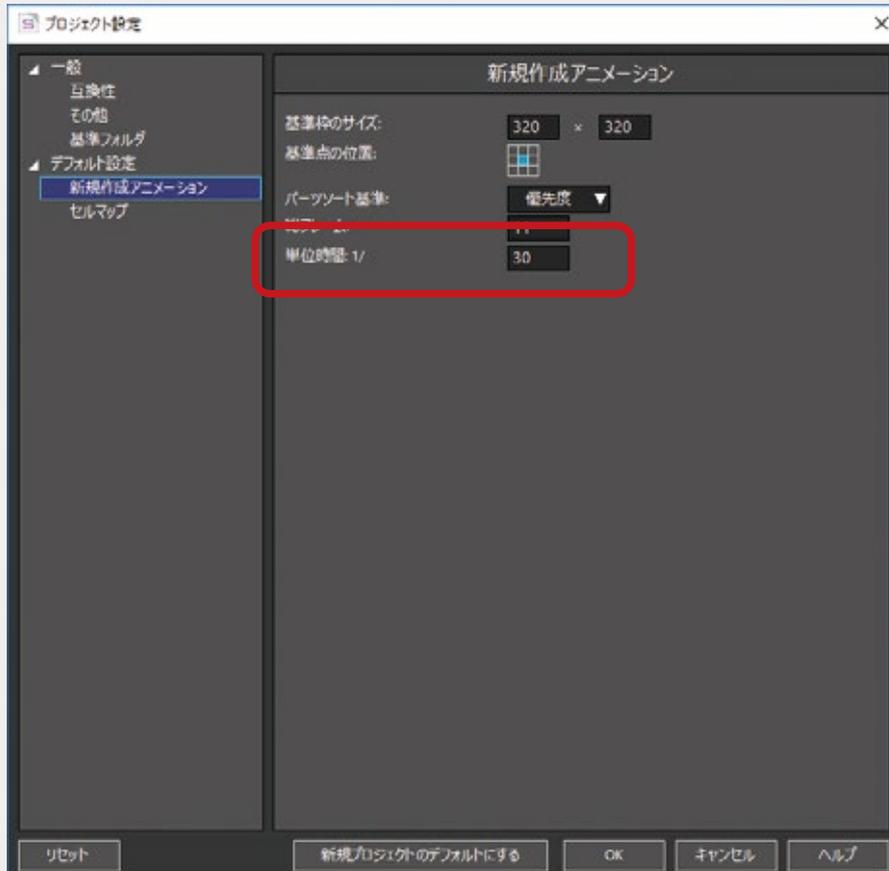


- 1 プルダウンボックスから選択するか任意の値を入力します

# HINT デフォルトの fps を設定する

プロジェクト設定の「新規作成アニメーション」でデフォルトの fps を設定することができます。「単位時間」の項目に fps を入力してください。

次回から新規作成するプロジェクトの fps が設定した値になります。



## パーツを動かしてみよう

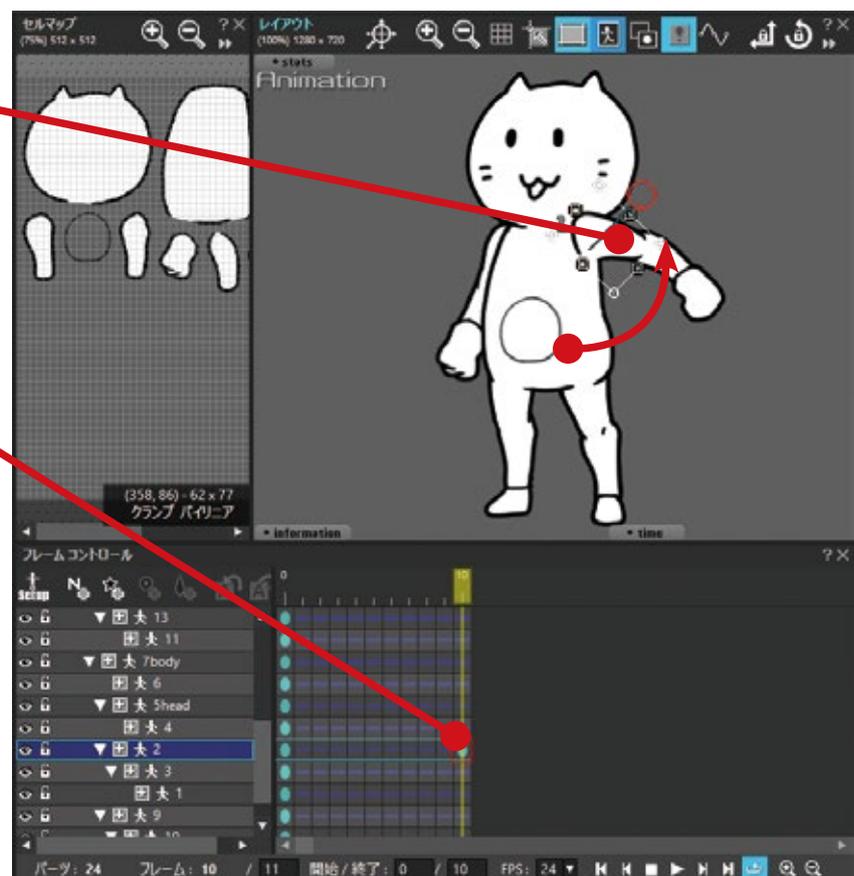
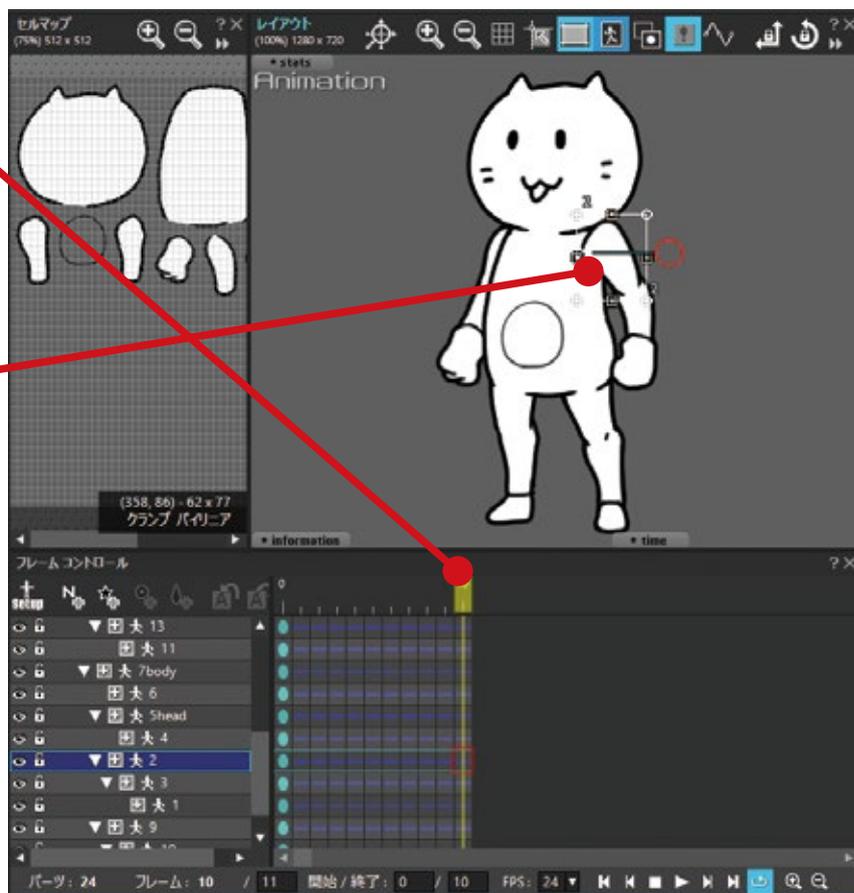
設定の確認が終わったら実際にパーツを動かしてみましょう。「アニメーションデータ」を選択します。

1 10フレーム目を選択します

2 腕のパーツを選択します

3 レイアウトウィンドウで  
パーツを動かします

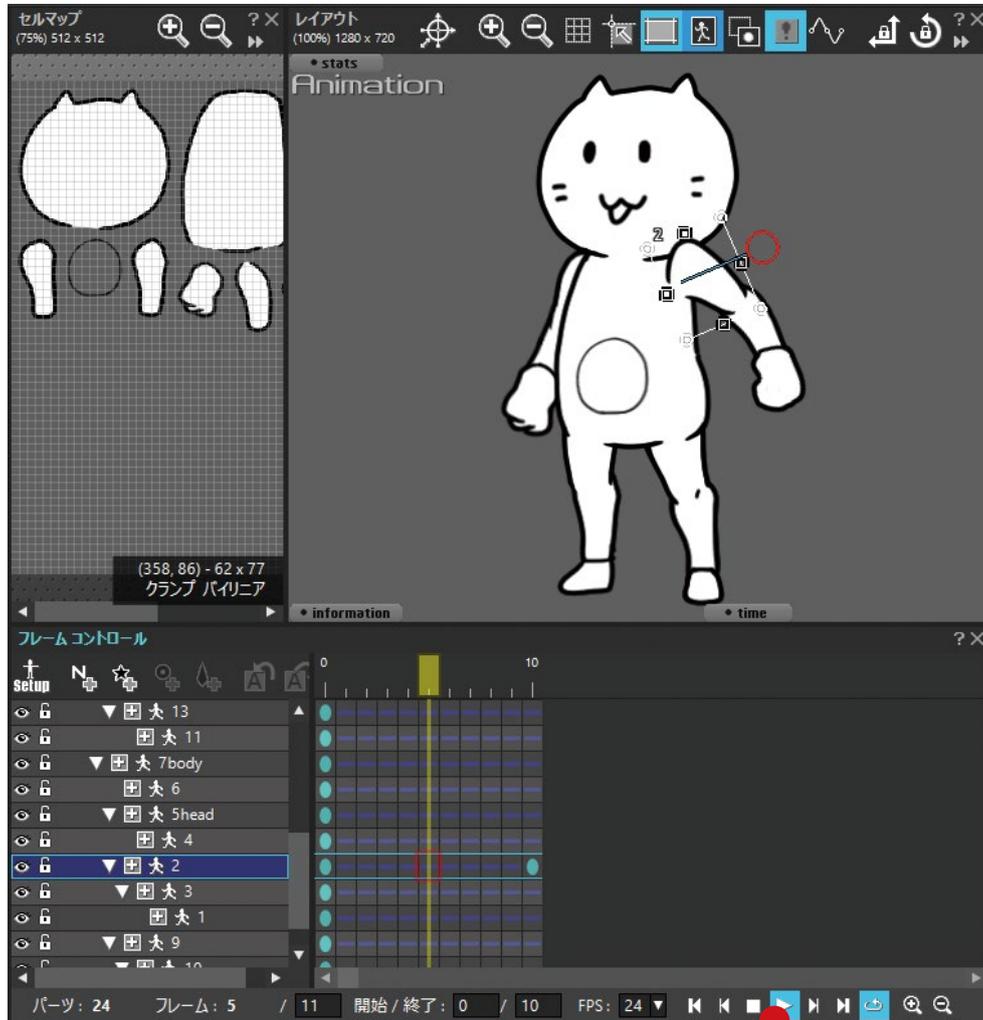
4 腕のパーツの10フレーム目に  
自動でキーが設定されました



## 動きを確認してみよう

設定が終わったら、フレームコントロールウィンドウの再生ボタンを押してみましょう。腕が上がるアニメが再生されます。

0 フレーム目から 10 フレーム目の間が補間されてアニメーションしているのが分かります。



5 再生ボタンを押すとアニメーションがプレビューされます

この作業を繰り返し、アニメーションを完成させていきます。

設定方法はわかったけれど実際にどのようにアニメーションさせるとよいか分からない……という方のために、本誌では巻末に『池田陽朗のアニメーション作成メイキング ~ 『てのひらの宇宙』はどのようにして作られたか』という記事を掲載しています。合わせてご一読ください。

# HINT 「キー」と「アトリビュート」

フレームコントロールウィンドウに表示される青い丸のことを「キー」と言います。

キーには様々な「アトリビュート」が格納されています。

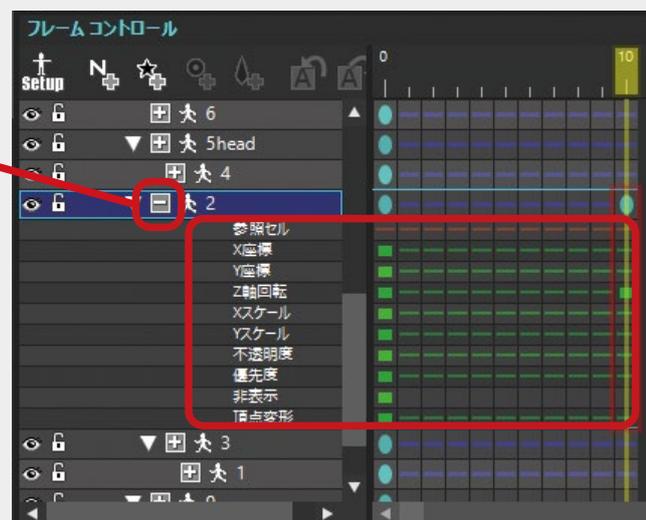
アトリビュートはいわば『情報(パラメータ)』です。例えばX座標やY座標といった位置情報や、不透明度といった情報のことを指します。

設定されたキーとキーの間のアトリビュートの値を自動で計算し、アニメーションさせてくれるのが「補間アニメーション」です。

この青い丸が「キー」です。  
この中にアトリビュートが  
格納されています  
そのアトリビュートに基づいて計算し、  
アニメーションを行います



**+** をクリックすると  
キーの中に含まれている「アトリビュート」  
を確認することができます  
含まれているアトリビュートは黄緑色の  
四角で表示されています



# HINT 補間方法を変更する

補間アニメーションはキーとキーの間を自動で補間して動かす方法です。基本的には「線形」という均等に補間をする「補間方法」が主ですが、等速で動くため、そのままではどうしても機械的に見えてしまいます。その際は補間方法を変更し、動きに変化をつけてみましょう。

フレームコントロールウィンドウのキー上で右クリックし、コンテキストメニューの「補間方法の変更」から変更を行ってください。



- 「線形」 基本となる補間方法です。均等に補間します。
- 「加速」 終点に向かって徐々に加速する動きになります。
- 「減速」 終点に向かって徐々に減速する動きになります。
- 「ベジェ」 キーの間を「ベジェ曲線」と呼ばれる曲線で補間します。  
カーブエディタ上のキーをアンカーポイントとし、ハンドルを操作して、任意のベジェ曲線を作成することができます。  
任意のなめらかなアニメーションを表現できます。
- 「エルミート」 キー間を三次曲線で補間します。三次曲線は「S字」のような変化をする曲線です。  
カーブエディタ上のキーをアンカーポイントとし、ハンドルを操作して、キーとキーの間のカーブを調整することができます。
- 「なし」 補間を行いません。アトリビュート値はそのキーになった段階で即座に切り替わります。

実際に補間方法を変更することでどのような動きをするか、詳しくはヘルプセンターをご確認ください。

<http://www.webtech.co.jp/help/ja/spritestudio/guide/tips/interpolation/>



# HINT 親子関係を設定する

腕や足といった多関節のパーツは一つのパーツを動かすと、それに連なるパーツもすべて動かす必要があります。パーツに親子関係を設定すると、親パーツの動きに子パーツを追従させて動かすことが可能です。



- 1 親にしたいパーツに子パーツをドラッグします  
今回は肩にあたるパーツを親パーツとし、その子パーツとして腕と手のパーツを紐づけます。

紐づけたいパーツにドラッグ



- 2 肩パーツに対して腕パーツが子として設定されました



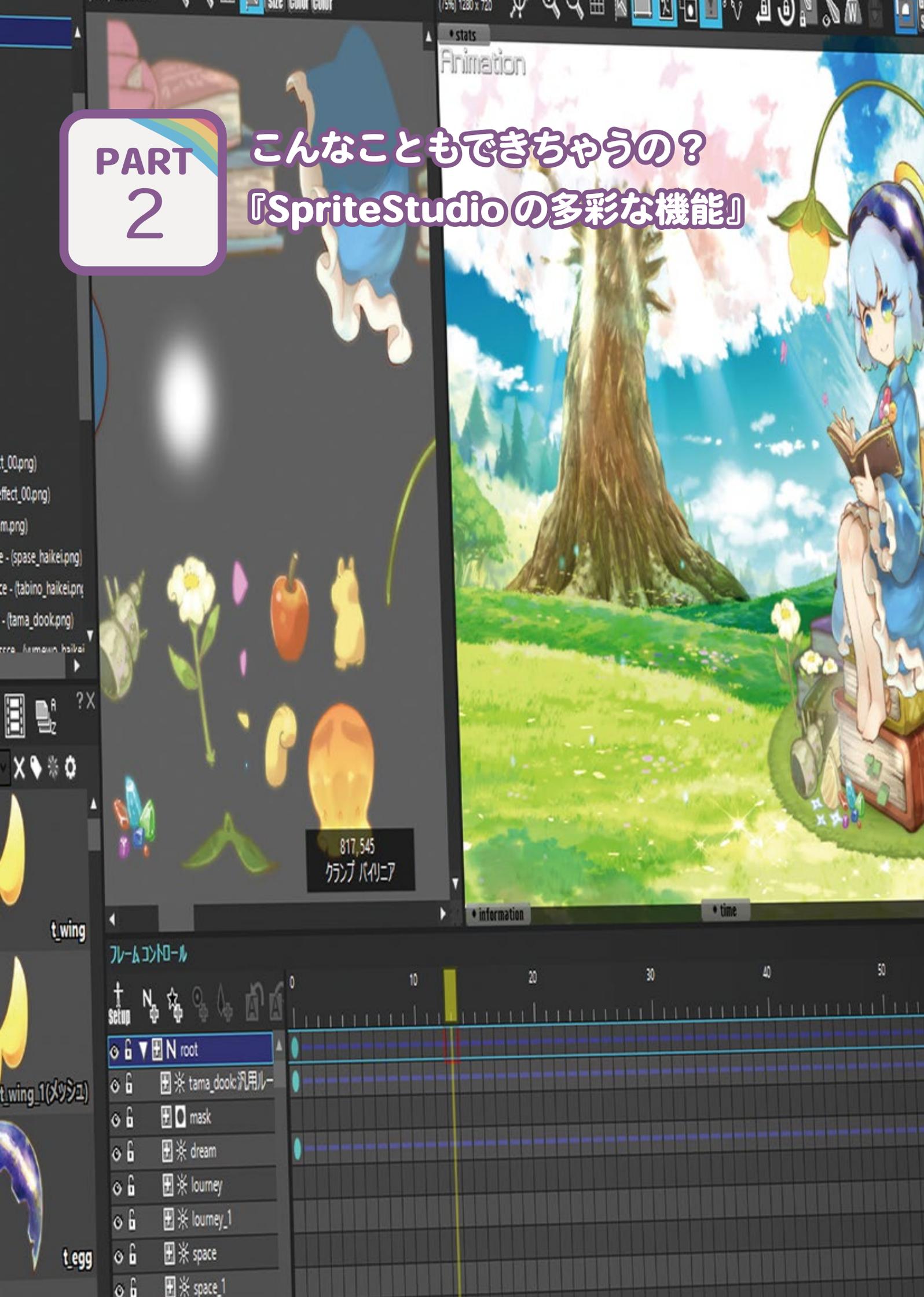
ほかにも「NULL パーツ」の効果的な使い方やテクニックをヘルプセンターで公開しています。

<http://www.webtech.co.jp/help/ja/spritestudio/guide/tips/nullparts/>



# PART 2

## こんなこともできちゃうの？ 『SpriteStudioの多彩な機能』





SpriteStudio の基本的な操作を覚えてアニメーションを作れるようになったあなた。でも、作れば作るほど「物足りない」「こんなことがしたい」という気持ちが出てくるように……。SpriteStudio はそんな「物足りなさ」を解消する様々な機能が存在します。この章では SpriteStudio の多彩な機能をご紹介します。

## 5 参照セルでコマアニメを作ろう

目パチ・ロパクといった絵柄が変わるアニメーションも、SpriteStudio なら簡単に作成することができます。「参照セル」を使った「コマアニメ」の作り方を解説します。

ここまで解説してきた「補間アニメーション」はパーツを移動させたり変形させることは可能ですが、絵柄を切り替える、いわゆる「コマアニメ」を作成することはできません。

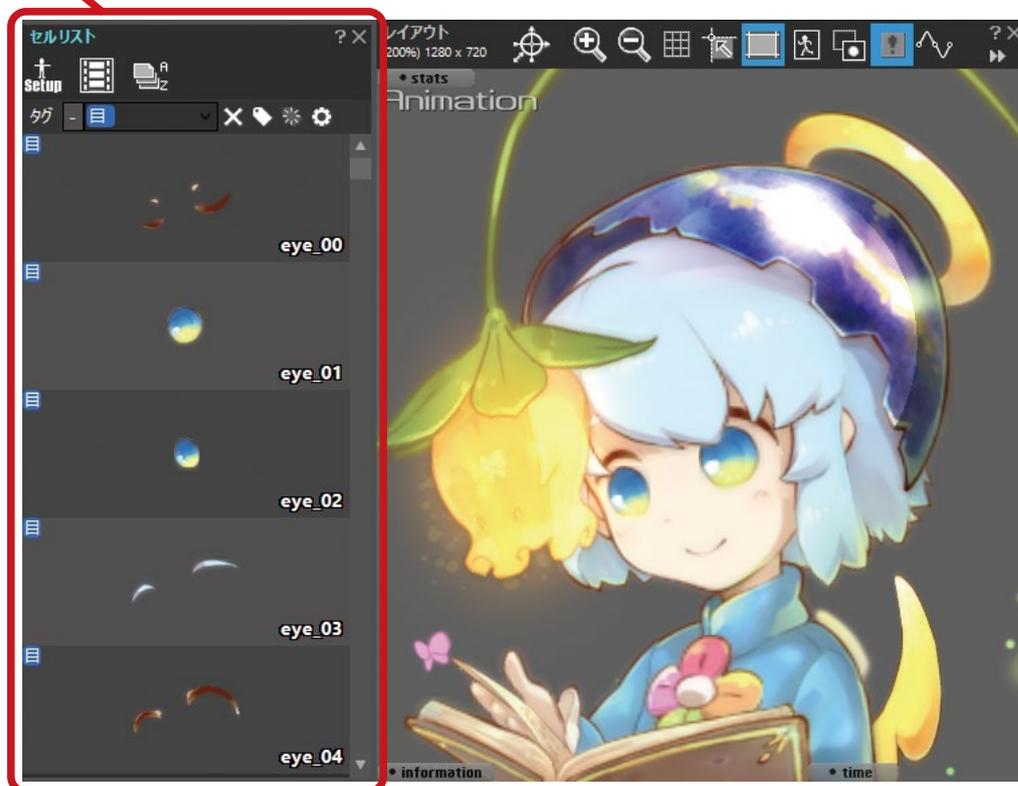
コマアニメを SpriteStudio で実現する場合「参照セル」を切り替えることで実現することが可能です

### 目パチアニメを作ってみよう

目パチのアニメは目自体の絵柄が変化するため、補間アニメーションでは表現が難しいアニメのひとつです。「参照セル」を切り替えて作ってみましょう。

まずは目パチに使う絵をセルマップに追加し、セルを作成します。

1 目パチに使う絵柄をセルに追加



アニメーションさせたい  
フレームを選択します。

アトリビュート「参照セル」を  
選択し、セルリストから  
目パチのパターンを選択します。

2 目パーツを選択



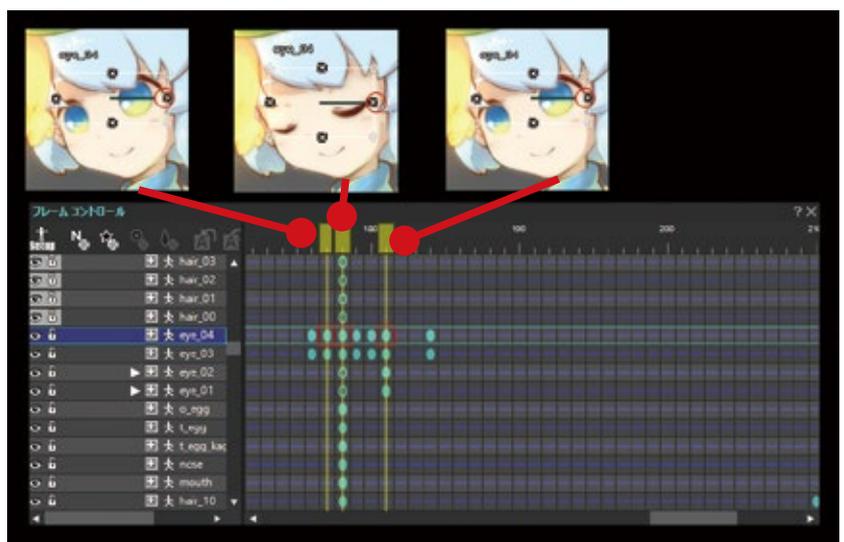
4 変更したいセルを選択

レイアウトウィンドウか  
フレームコントロールウィンドウ上で  
目パーツを選択します。



5 このフレームまできたら目パーツの  
参照セルが変化するようになりました

各フレームに参照セルを  
設定する作業を繰り返します。  
今回はこのようになっています。

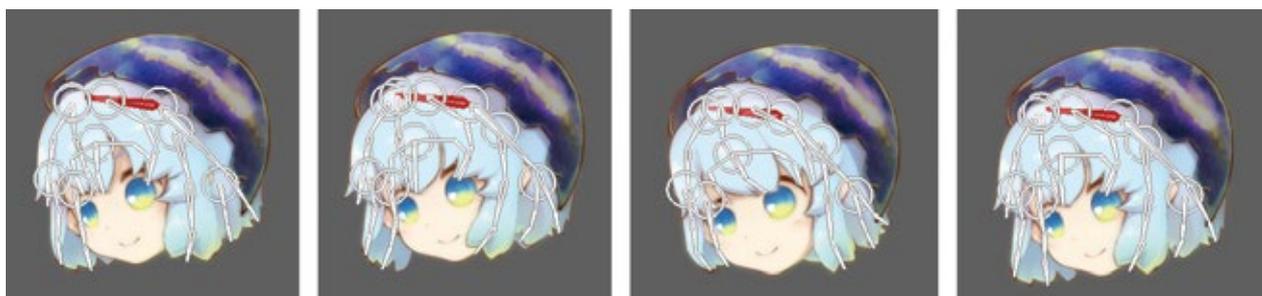


これで目パチアニメの完成です。違和感がある場合は変化させるフレームの位置を調整してください。

## 6 メッシュ・ボーン機能で柔らかい動きを作ろう

布をはためかせたり、髪をなびかせるようなアニメーションは、SpriteStudio Ver.6 で新たに実装したメッシュ・ボーン機能を使うことで簡単に実現することができます

通常、補間アニメーションでは布をはためかせたり、髪をなびかせるような、柔らかなアニメーションを実現するには、大変な労力が必要です。  
ですが、SpriteStudio Ver.6 で新たに実装したメッシュ・ボーン機能を使うことで、柔らかな表現や立体的なアニメーションを簡単に作成できます。



### セットアップデータに切り替えよう

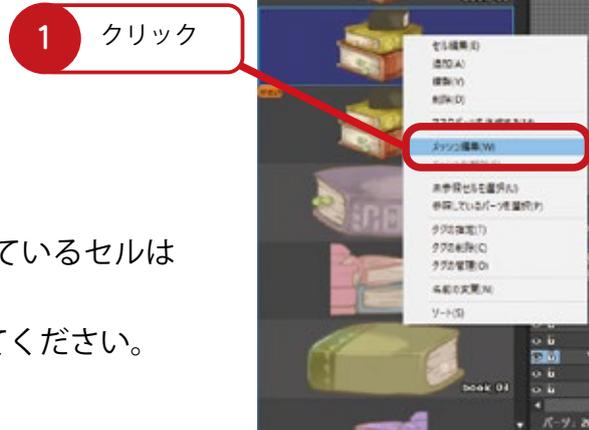
メッシュ・ボーン機能を活用するには、セルをメッシュにしないてはいけません。  
まずはセットアップデータに切り替えます。



フレームコントロールウィンドウや、セルリストウィンドウのアイコンをクリックしても切り替えることができます。

## セルをメッシュ分割しよう

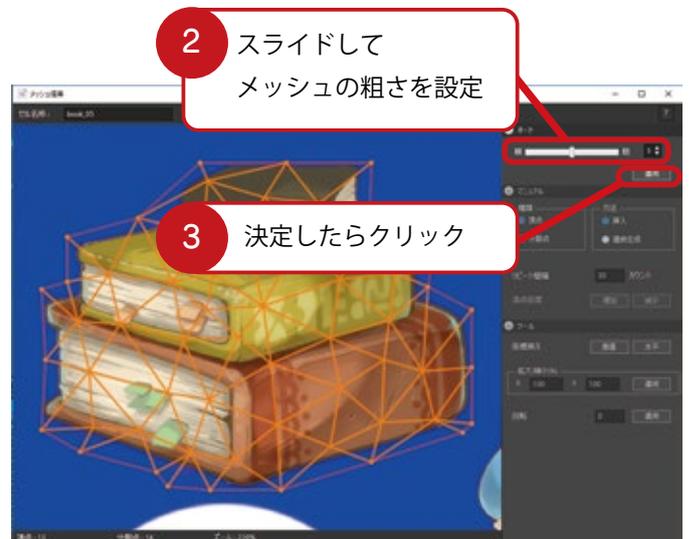
セットアップデータの状態で、メッシュにしたいセルを右クリックして「メッシュ編集」を選択します。



すでにアニメーションやエフェクトで参照されているセルはメッシュ編集することができません。その場合はそのセルを複製し、メッシュ編集してください。

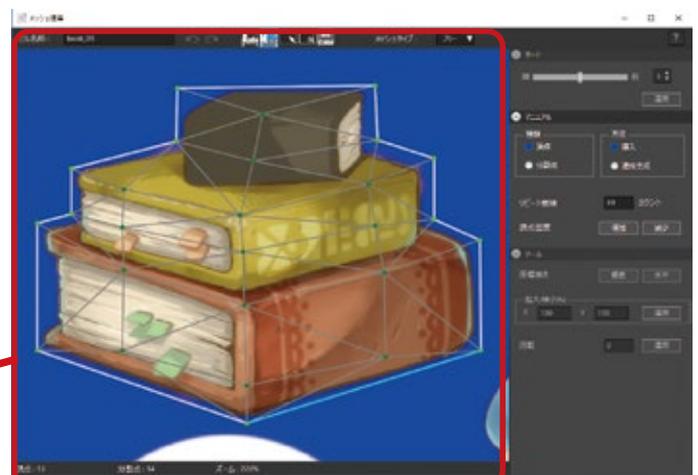
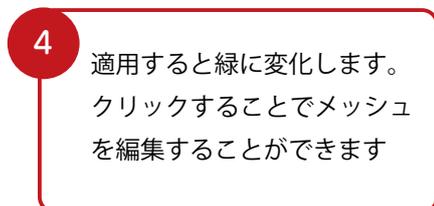
開くと自動でメッシュが生成されます。「粗」側にスライダーを動かすとメッシュは粗く「細」側にスライダーを動かすとメッシュはより細くなります。

メッシュの粗さを決めたら「適用」を押してください。

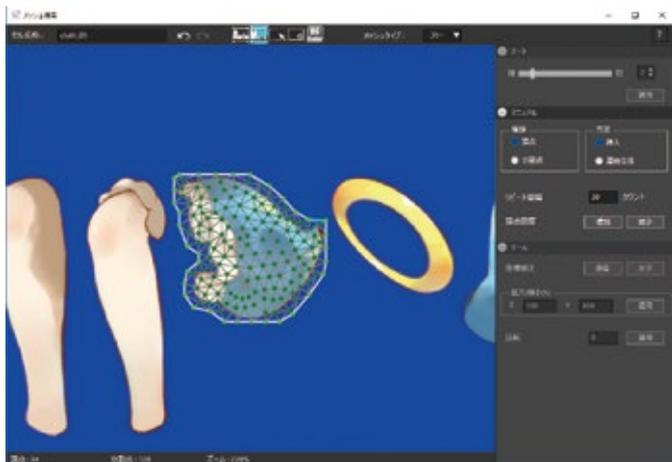


適用するとメッシュが確定します。問題なければそのままメッシュ編集ウィンドウを閉じて保存をしてください。

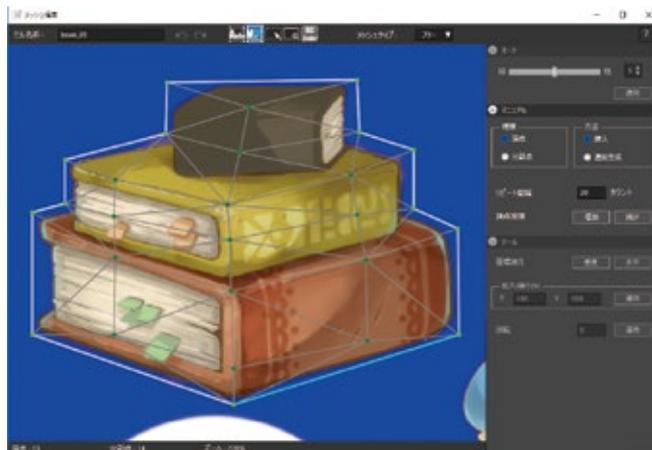
手で頂点や分割点を編集して、メッシュを作成することもできます。



髪の毛や布といった柔らかな動きであれば、細かくすることでより滑らかに動きます。



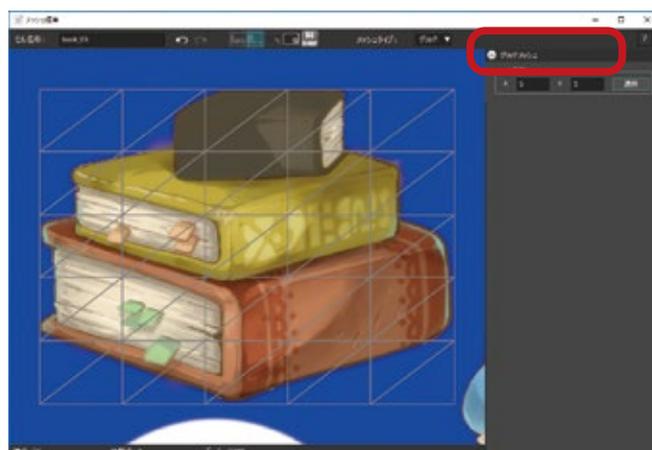
本や箱といった「面」で動くものを動かしたい場合、面を意識した分割をすることで、イラストを立体的に動かすことができます。



ただし、メッシュを多く分割しすぎると、その分、動作やデータが重たくなってしまいます。使用用途と動かしたいパーツがどのようなものかを確認し、適切な分割数を設定しましょう。

## 分割タイプを変更してみよう

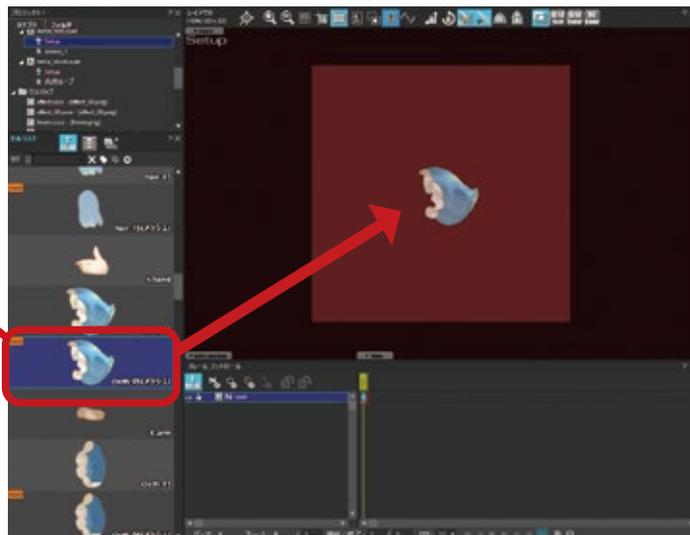
分割タイプを「グリッド」に変更することで、セルの大きさに合わせて三角形で分割します。



## ジョイントパーツとメッシュパーツを設定しよう

メッシュにしたセルを  
レイアウトウィンドウに  
ドラッグドロップします。

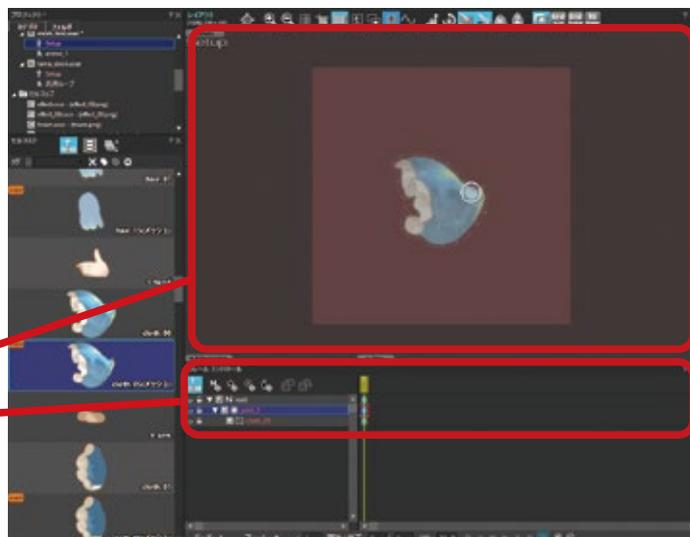
- 1 メッシュセルを  
ドラッグドロップします



これでメッシュセルと  
ジョイントパーツの配置は完了です。

ジョイントパーツの位置は  
セルの原点の位置に配置されます。

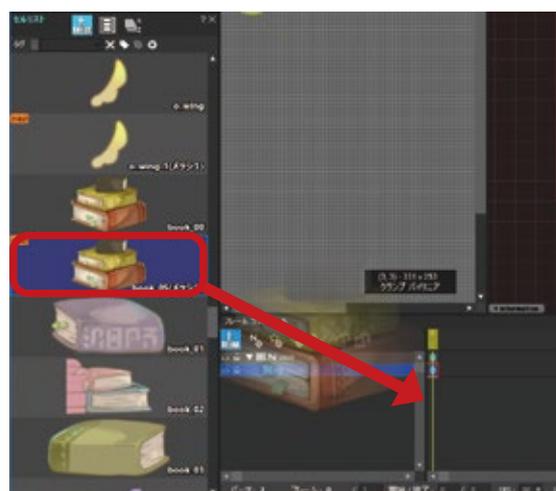
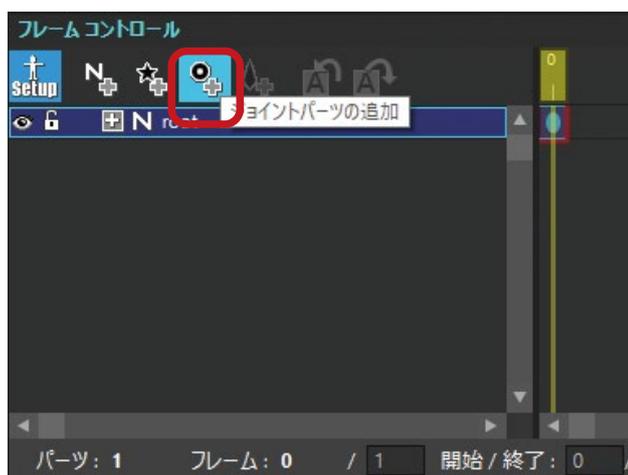
- 2 メッシュパーツとジョイント  
パーツが配置されました



先にジョイントパーツを配置してから、メッシュパーツを配置することもできます。

フレームコントロールウィンドウで  
「ジョイントパーツを追加」をクリックします。

ジョイントパーツの子パーツとして  
メッシュセルを追加します。

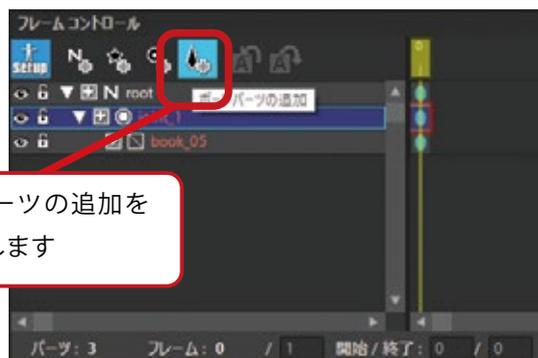


## ボーンパーツを設定しよう

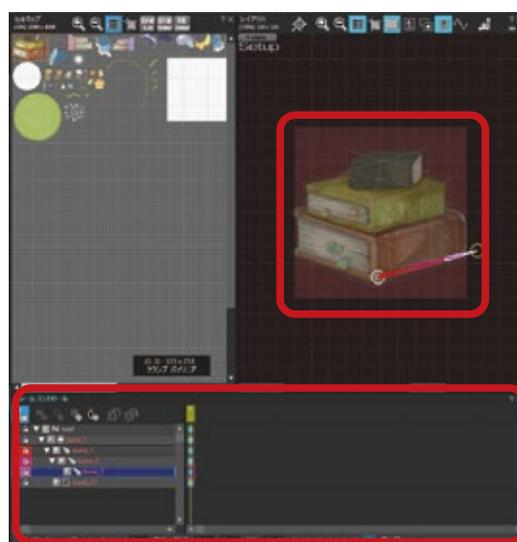
ボーンパーツの追加をクリックします。

なお、ボーンパーツはジョイントパーツかボーンパーツに紐づく形でのみ追加できます。

ボーンパーツの追加をクリックします

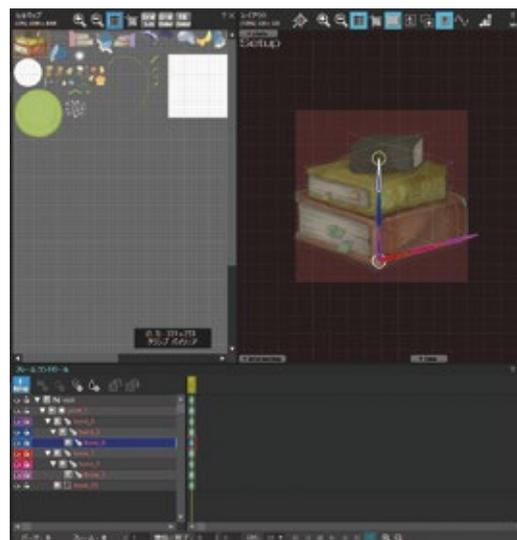


ボーンパーツが追加されました。



ボーンパーツはジョイントパーツから分岐させることができます。

また、メッシュパーツに複数のボーンパーツを配置することも可能です。



# アニメーションを設定しよう

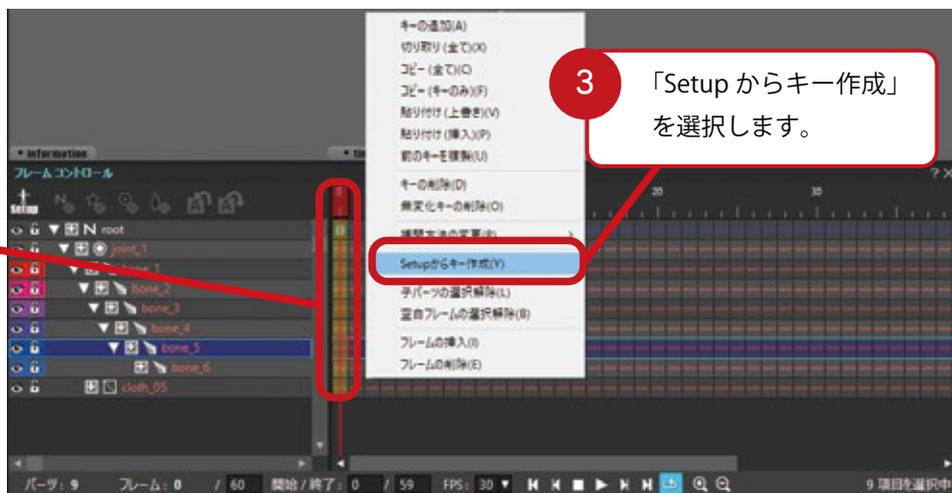
まずはセットアップデータをアニメーションデータに変更します。

- 1 レイアウトウィンドウの Setup をクリックすると Animation に切り替わります



0 フレーム目を選択し  
右クリックをして「setup から  
キーを作成」を選択します。

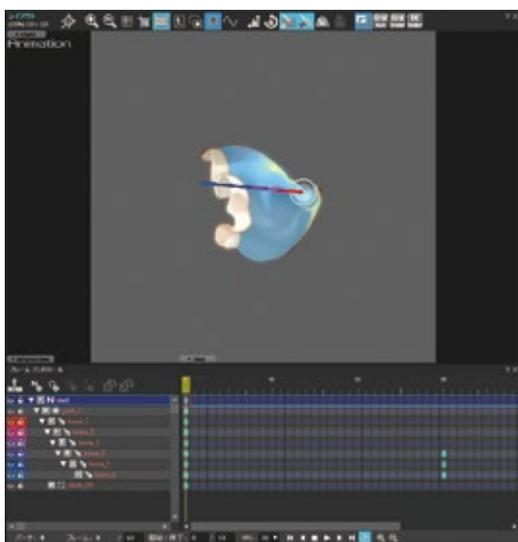
- 2 Shift + クリックで  
範囲選択します



- 3 「Setup からキー作成」  
を選択します。

セットアップデータで作成したキーが 0 フレーム目に反映されました。

通常パーツと同様にボーンパーツを編集することで、メッシュパーツにアニメーションを設定できます。



- 4 ボーンパーツを動かすと  
それに応じてメッシュが  
アニメーションします

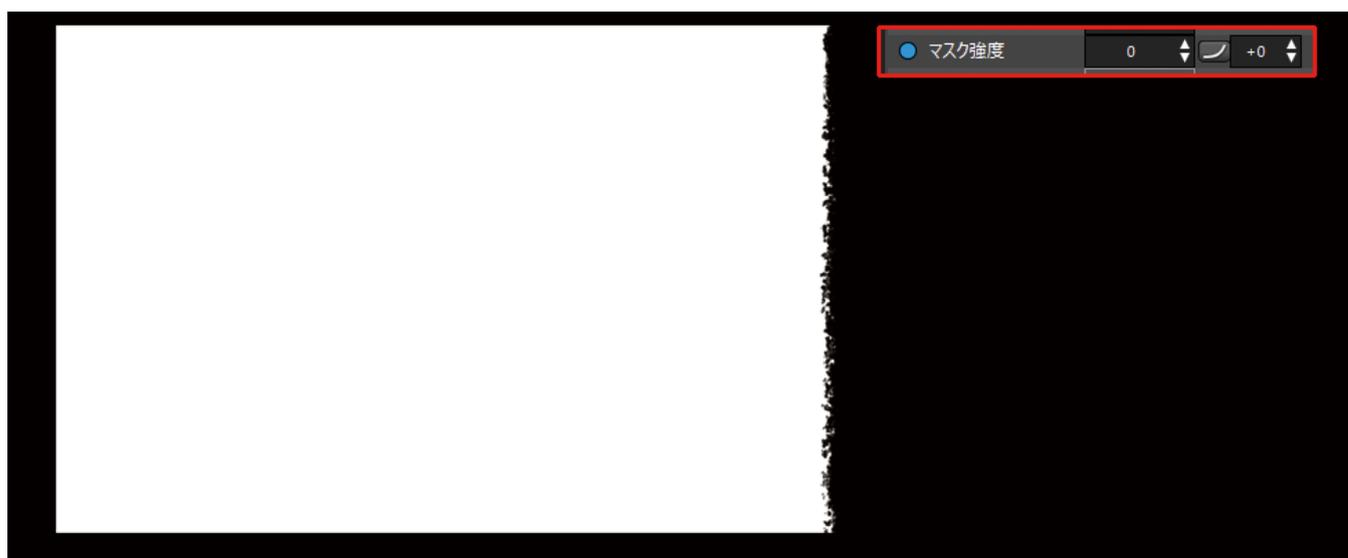
## 7 マスク機能で凝った演出を作ってみよう

ワイプやカットインなど、もっと凝った演出を作りたい……。そんな時はマスク機能を使ってみましょう。SpriteStudioのマスク機能はとても簡単に複雑な演出を実現してくれます。

### マスク機能のしくみ

SpriteStudioのマスク機能は、セルの「アルファチャンネルの値」とアトリビュートの「マスク強度」により制御されています。

マスクパーツに設定されたセルの「アルファチャンネルの値」と「マスク強度」の合計が255よりも大きくなると、マスクパーツの下のパーツが表示されます。うまくパーツを作ることで「キャラクターがじわじわ消える」などの演出が簡単に作れます。



セルのアルファチャンネルの値



アトリビュート「マスク強度」の値

合計値が 255 を超えるとマスクが有効になる

### HINT マスクパーツを作るコツ

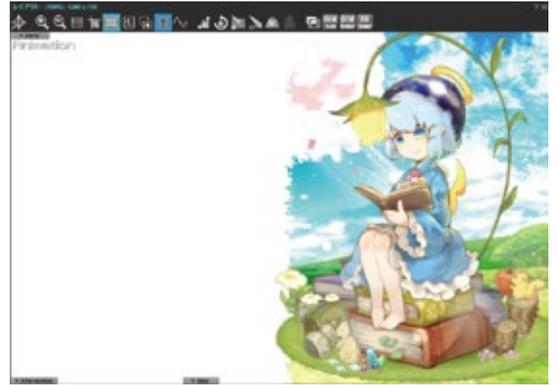
マスクの境界にアンチエイリアスはかかりません。  
できるだけ画面のピクセルに合わせたマスク画像を作成したほうが良いでしょう。  
(拡大するとピクセルが目立ってしまいます)

## 実例をみてみよう

右図のようなパーツを作ります。



通常パーツとして載せるとただの白い板ですが……



マスクパーツとして配置すると、  
パーツがマスクされて  
奥の背景パーツが表示されました。



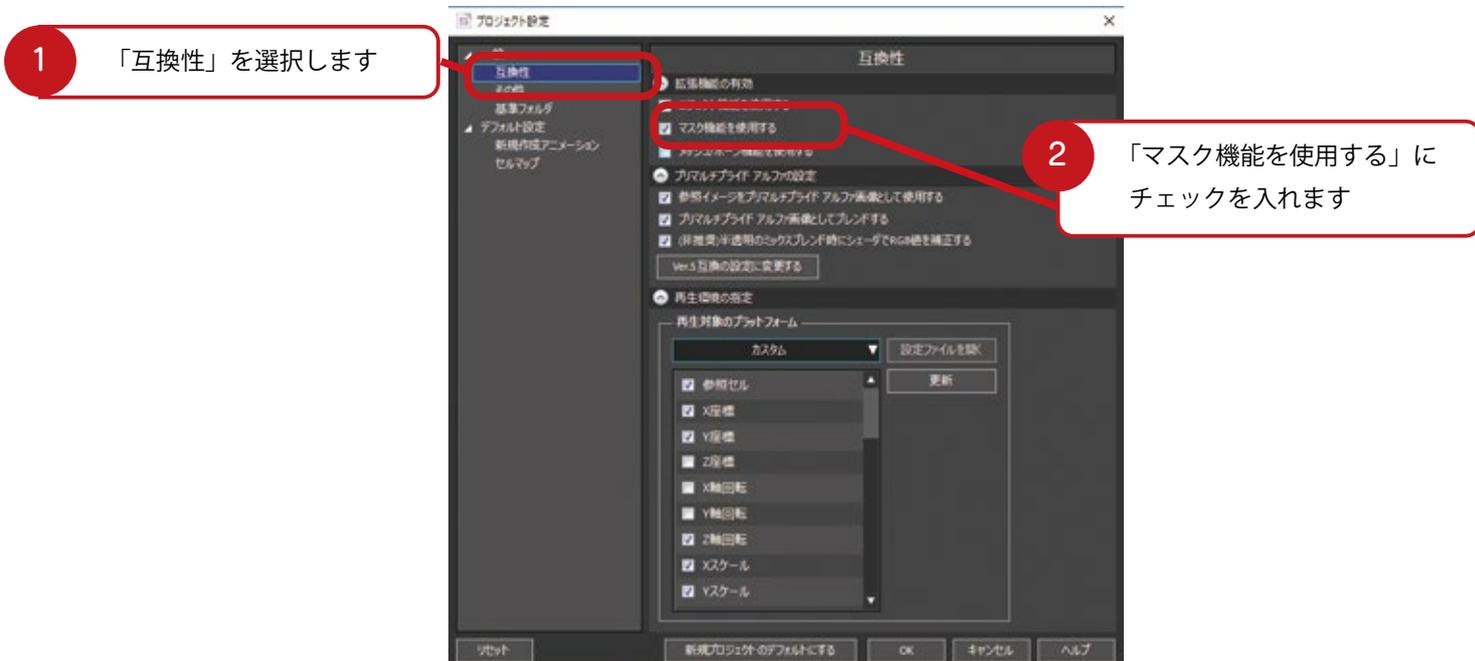
うまく活用することで  
カットイン演出など  
さまざまな演出を作ることが可能です。



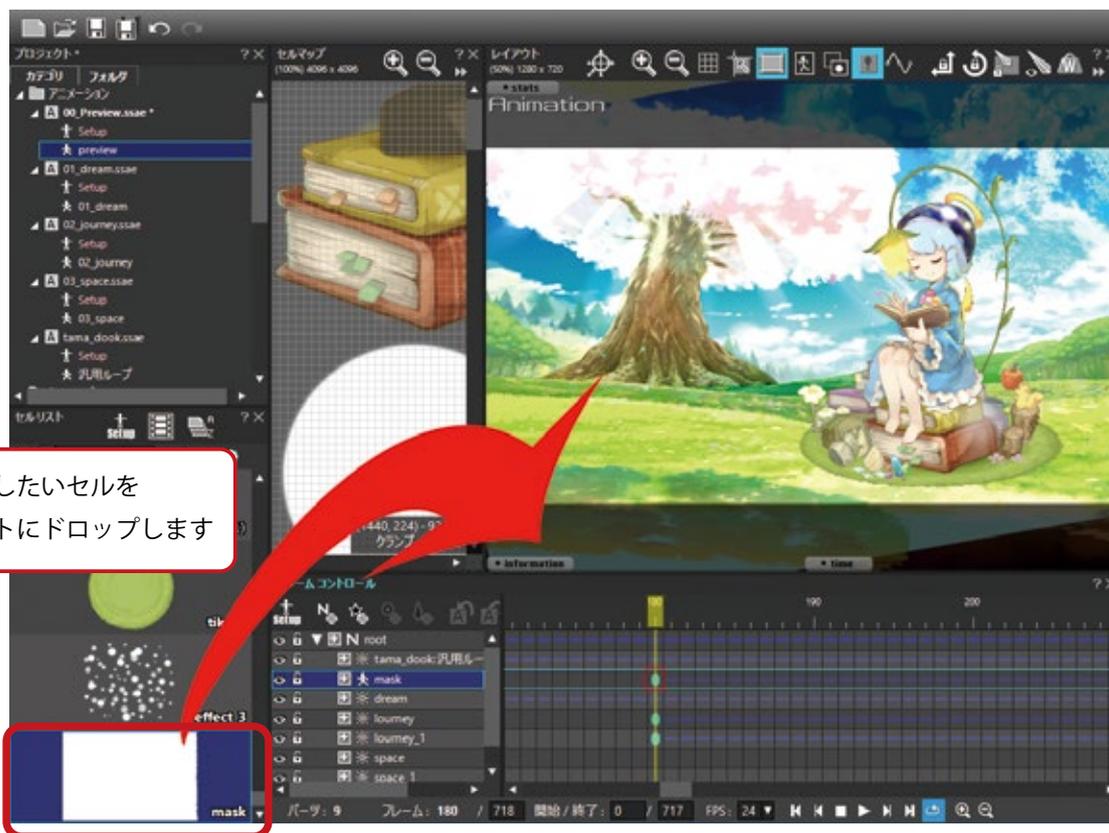
## プロジェクト設定の「互換性」を確認しよう

マスク機能を利用する際は「プロジェクトの設定」の互換性の項目を確認してください。「マスク機能を使用する」にチェックが入っていないとマスク機能を利用することができません。

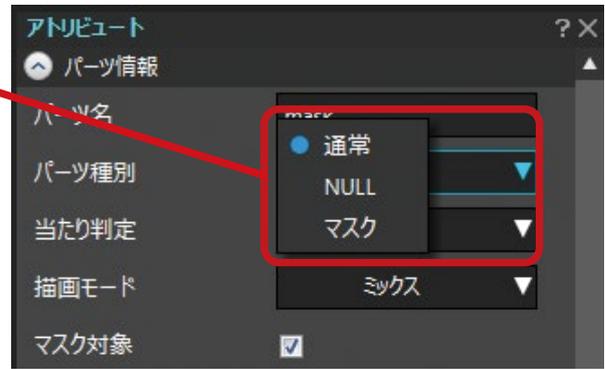
「ファイル」メニューの「プロジェクトの設定」から「互換性」を選択し、「マスク機能を使用する」にチェックが入っているかどうかを確認してください。



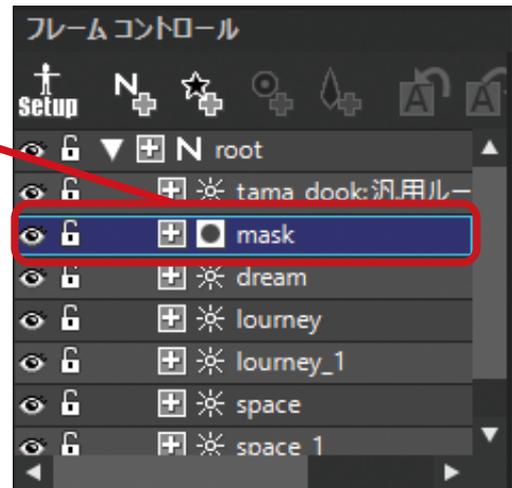
## マスク機能を使ってワイプ演出を作ってみよう



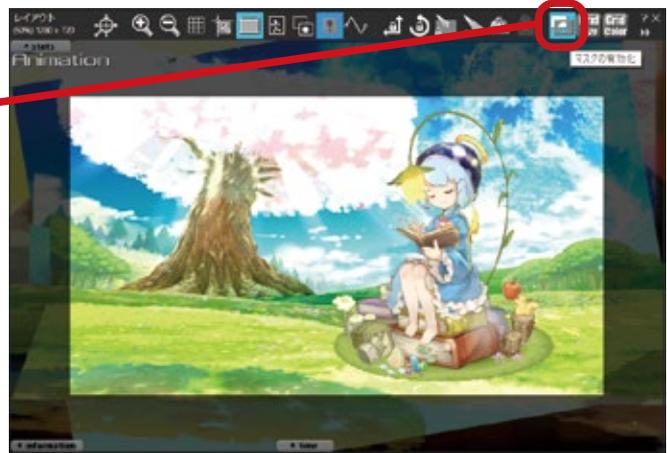
2 パーツ種別を「マスク」に変更します



3 パーツが「マスクパーツ」に変更されました



4 「マスクの有効化」をONにします



6 マスクパーツの下にあるパーツが表示されるようになりました  
マスクパーツを横断させることで  
ワイプ演出は完成です

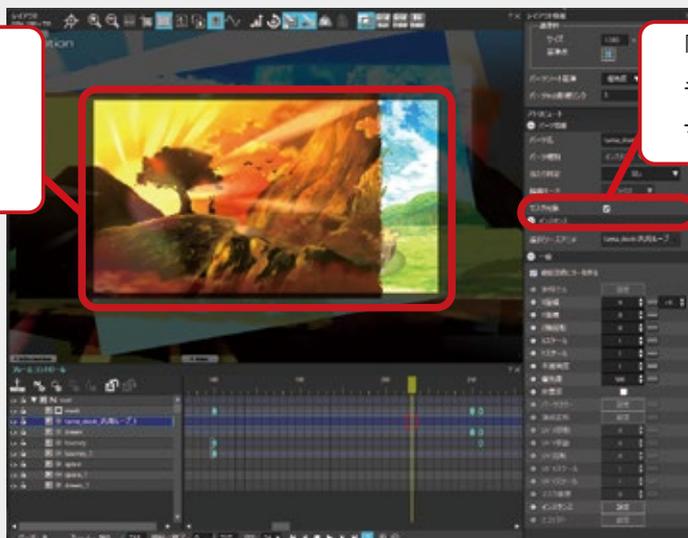


5 「マスク強度」の値を設定します

# HINT マスクの対象にするパーツ、しないパーツ

特定のパーツだけマスクで抜けないようにしたい時は「マスク対象」のチェックを外してください。

背景だけをマスクしたいのに  
女の子までマスクされて  
消えてしまいました



「マスク対象」に  
チェックが入っているため  
マスクされている状況です

背景だけがマスクされるよう  
になりました



女の子のパーツの  
「マスク対象」の  
チェックを外します

# HINT 各パーツをマスクしたらどうなるの？

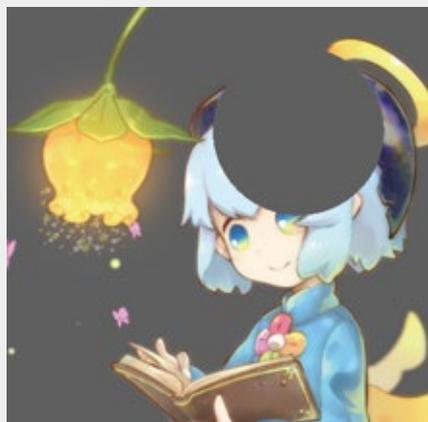
通常パーツ、メッシュパーツ……SpriteStudioには様々なパーツが存在します。各パーツにマスクをかけたとき、どのように表示されるか確認しましょう。

## 通常パーツ

### メッシュパーツ

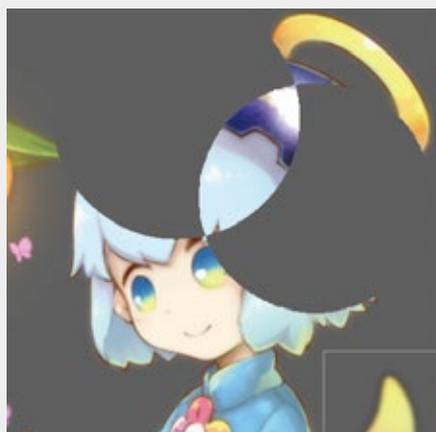
### エフェクトパーツ

マスクパーツの下に置いた各パーツはマスクされ、見えなくなります。



## マスクパーツ

重なった部分のマスク効果を打ち消します。



**インスタンスパーツ** ソースアニメーション内のマスクパーツは無視されます。

### マスク対象を有効にしている場合



ソースアニメーションでの各パーツの「マスク対象」の設定に則ってマスク対象となります。

### マスク対象を無効にしている場合



ソースアニメーションの各パーツの「マスク対象」の設定に関わらず、すべてがマスクの対象外になります。

## 8 エフェクト機能でパーティクルエフェクトを作ろう

演出でよく見られる「小さなキラキラが散る」エフェクト。あのキラキラをひとつずつパーツで登録して動きを調整して……考えるだけで気が遠くなるような作業です。

SpriteStudioのエフェクト機能を使えば、パラメータを設定するだけで簡単に作れます。

魔法やモンスターが消滅するときの演出でよく見られる「小さなキラキラが散る」エフェクト。あのようなエフェクトは「パーティクルエフェクト」といいます。

「パーティクル」を不規則に動かすエフェクトで、不規則な形状・動きをする火の粉や煙を表現するために生まれました。

SpriteStudioのエフェクト機能を使えば、パラメータを設定するだけでひとつずつパーツの調整をしなくても、簡単にパーティクルエフェクトが作れます。

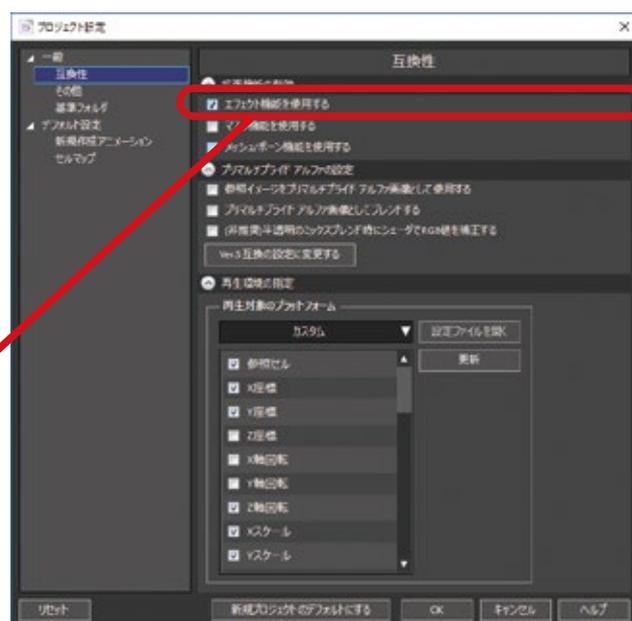
### プロジェクト設定の「互換性」を確認しよう

エフェクト機能を利用する際は「プロジェクト設定」の互換性の項目を確認してください。

「エフェクト機能を使用する」にチェックが入っていないとエフェクト機能を利用することができません。

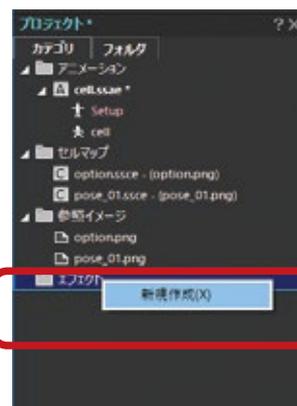
「ファイル」メニューの「プロジェクト設定」から「互換性」を選択し、「エフェクト機能を使用する」にチェックが入っているかどうかを確認してください。

- 1 「エフェクト機能を使用する」にチェックを入れます



プロジェクトウィンドウに「エフェクト」の項目が表示され、エフェクトを新規作成できるようになります。

- 2 「エフェクト」で右クリックし「新規作成」を選択



## エフェクトにしたいパーティクルをセルに登録しよう

エフェクトを作成するには「パーティクル」が必要です。パーティクルはエフェクトを構成する粒子のことをさします。

ssce を作成し、パーティクルに設定したい画像が含まれているテクスチャを参照イメージに設定します。

ssce を作成した後、セルを作成し、セルリストに登録します。



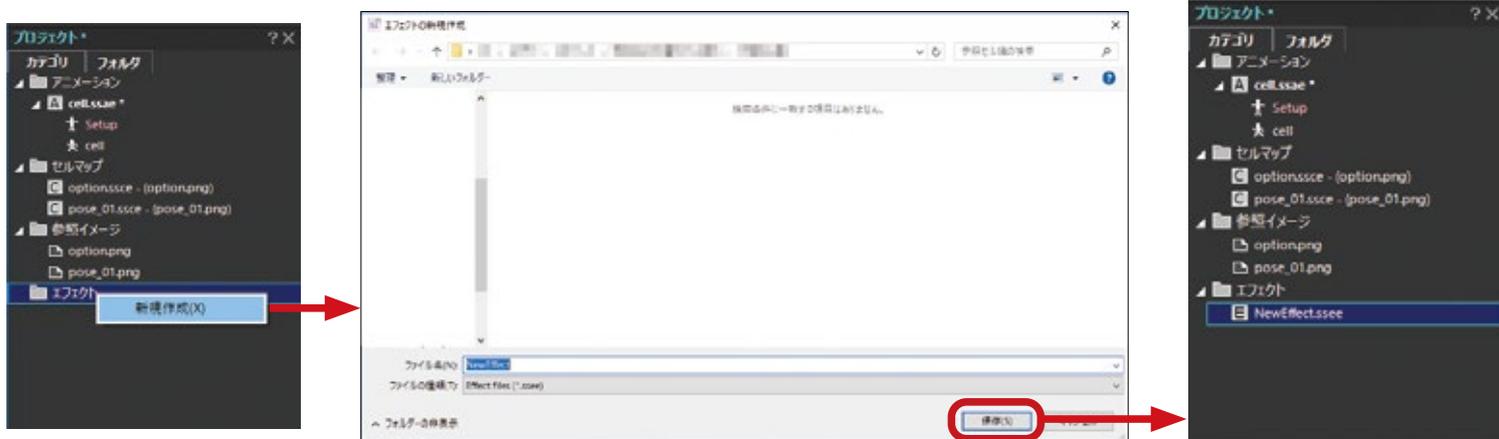
3 エフェクトに使用するパーティクルの画像をセルとして登録

## ssee(エフェクトファイル) を作ろう

エフェクトは ssee(エフェクトファイル) で管理しています。そのため、ssae・ssce とは別に、新たに ssee を作成しなければいけません。

下記のいずれかの方法で新規 ssee を作成してください。

1. メインメニューの「プロジェクト」から「新規ファイルの追加」→「エフェクト」を選ぶ
2. プロジェクトツリーのエフェクトカテゴリのコンテキストメニュー→「新規作成」を選ぶ



# エフェクトエディタを使ってみよう

既にエフェクトファイルを作成している場合は、編集を行う ssee を選択し、ダブルクリックするか、ssee のコンテキストメニューの「開く」より開くことができます。

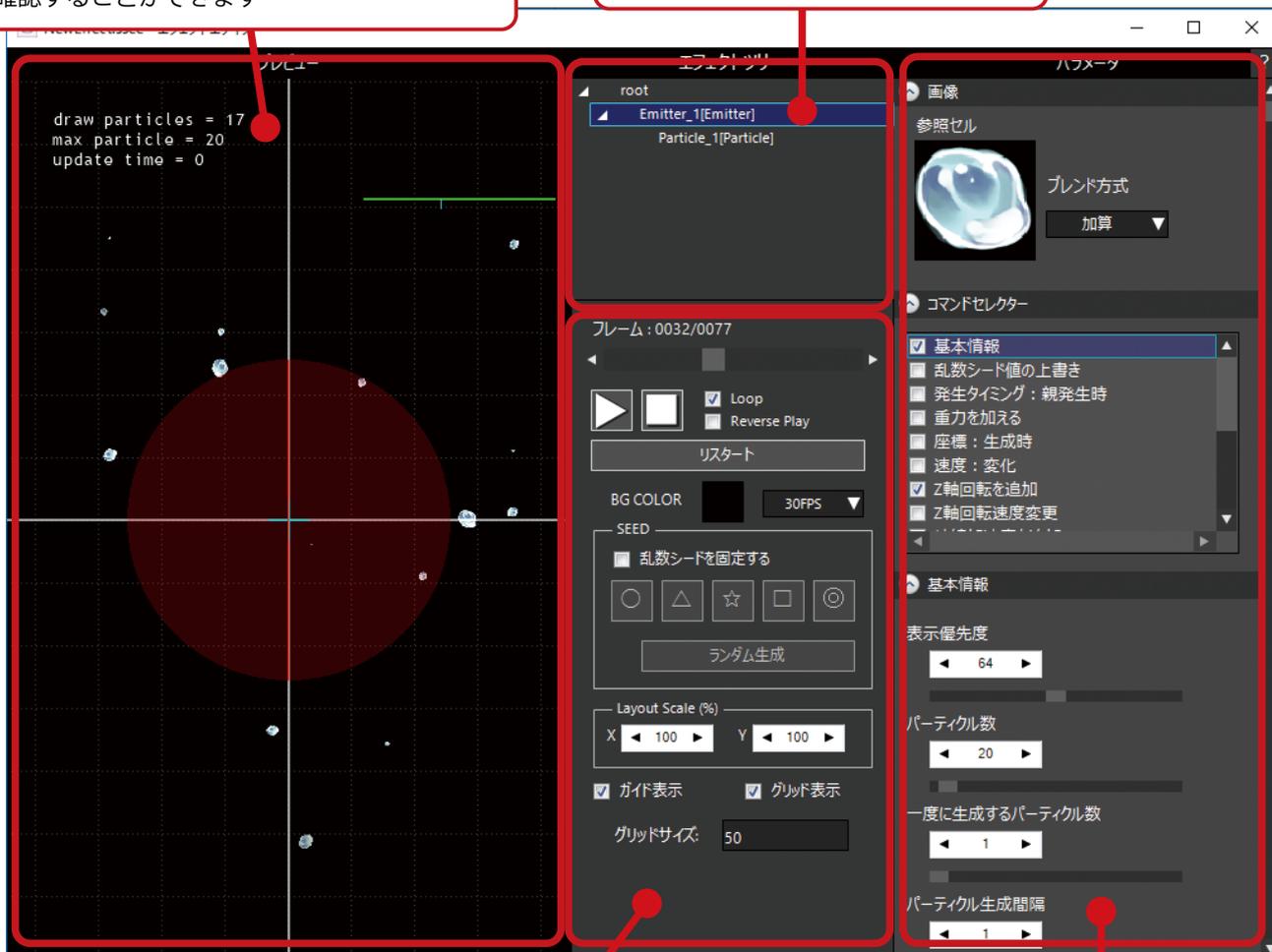
ssee を開くとエフェクトエディタが表示されます。

## 1 プレビュー

現在編集しているエフェクトがどのようにアニメーションするか確認することができます

## 2 エフェクトツリー

エミッタの選択・追加・削除を行うことができます



## 3 コントロールパネル

プレビューの再生制御を行います

## 4 パラメータエディタ

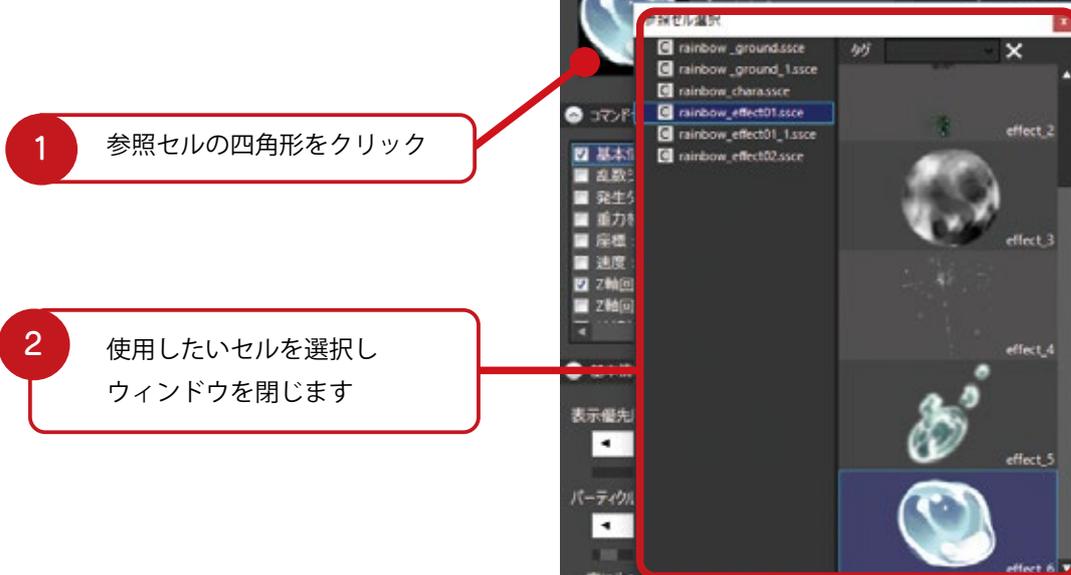
使用するパーティクル画像の選択やエフェクトの動きを設定します

# HINT パーティクルとエミッタ

パーティクルとエミッタ。この二つはパーティクルエフェクトを作る上で欠かせない要素です。エミッタは「パーティクルの射出場所」を指します。パーティクルは「エフェクトで表示される粒子」を指します。

## パーティクルに使用する画像（セル）を選択

パーティクルに使用する画像を選択します。  
パラメータの黒い四角をクリックすると、「参照セル選択」ダイアログが表示されます。  
ここからパーティクルとして使用したいセルを選択します。



## コマンドセクターで動きを設定する

エフェクトの動きは「コマンドセクター」の各種コマンドを使用して設定します。

1

### 「基本情報」で設定を行う

コマンドセクターの一番上にある「基本情報」は、そのエミッタの基本となる情報を設定する項目となっており、選択を外すことはできません。

パーティクル数や生成間隔等を設定することができます。

2

### 実際にパラメータを設定する

基本情報以外のパラメータは「どのようにパーティクルを動かすか」のパラメータです。

たとえば「Z軸回転を追加」にチェックを入れ、パラメータを設定すると、パーティクルがくるくる回転するアニメーションに変化します。



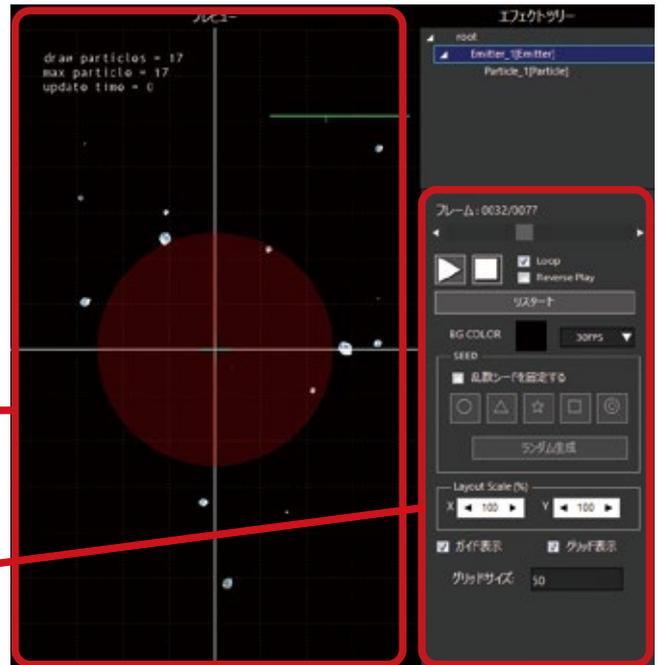
## プレビューで確認する

エフェクトエディタのプレビュー部分で、設定したエフェクトの動きを確認することができます。

完成したらエフェクトエディタを閉じます。  
作ったエフェクトをアニメーションに組み込んでいきます。

1 現在の設定での  
アニメーションを確認できます

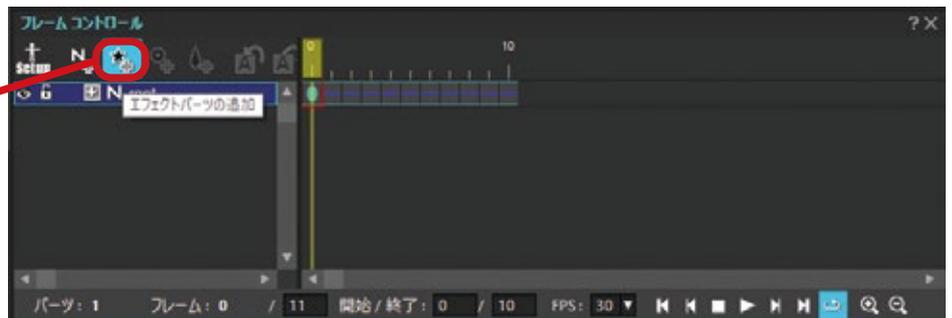
2 再生・停止やループ再生などは  
コントロールパネルで行います



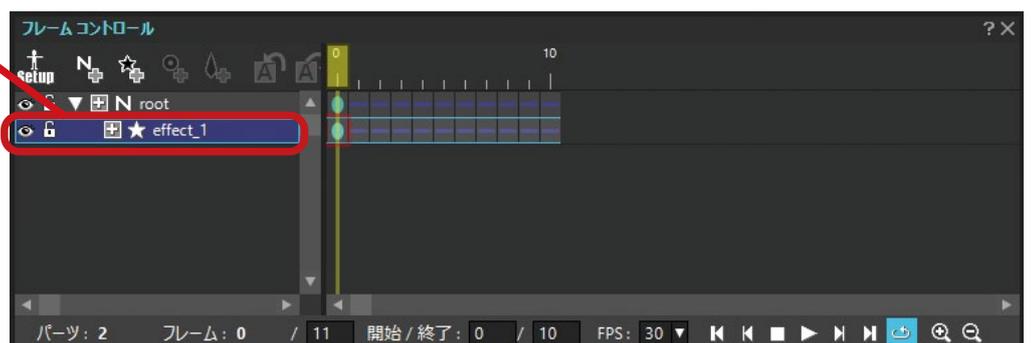
## 作ったエフェクトを「エフェクトパーツ」として組み込む

フレームコントロールウィンドウで「エフェクトパーツ」をアニメーションに追加します。  
エフェクトパーツとは ssee を参照するパーツです。

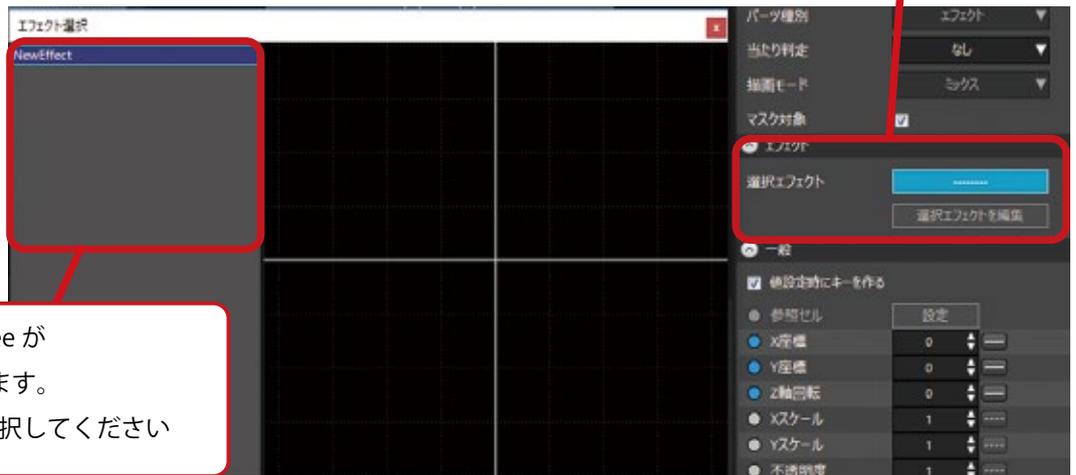
1 「エフェクトパーツの追加」  
ボタンをクリックします



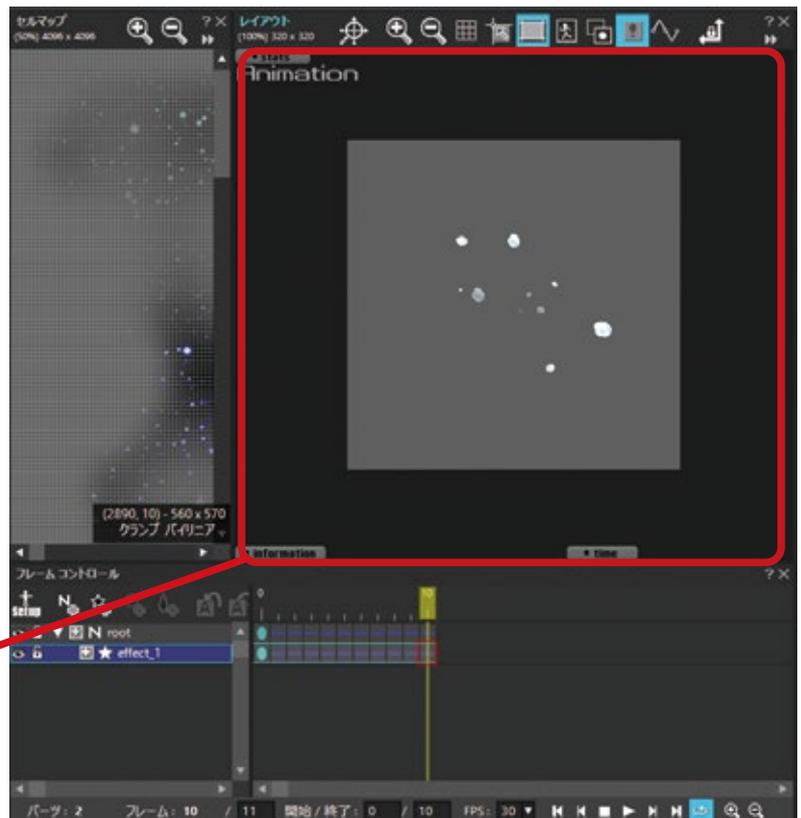
2 エフェクトパーツが  
追加されました



3 「エフェクト」欄で  
参照する ssee を選択します



4 sspj に含まれる ssee が  
リストで表示されます。  
使用する ssee を選択してください



5 エフェクトパーツで選択した ssee が  
アニメーションに表示されました

コマンドセクターの各種パラメータについて  
詳しくはヘルプセンターをご覧ください



[http://www.webtech.co.jp/help/ja/spritestudio/guide/window6/other/effecteditor/#effect\\_editor\\_command](http://www.webtech.co.jp/help/ja/spritestudio/guide/window6/other/effecteditor/#effect_editor_command)

## 9

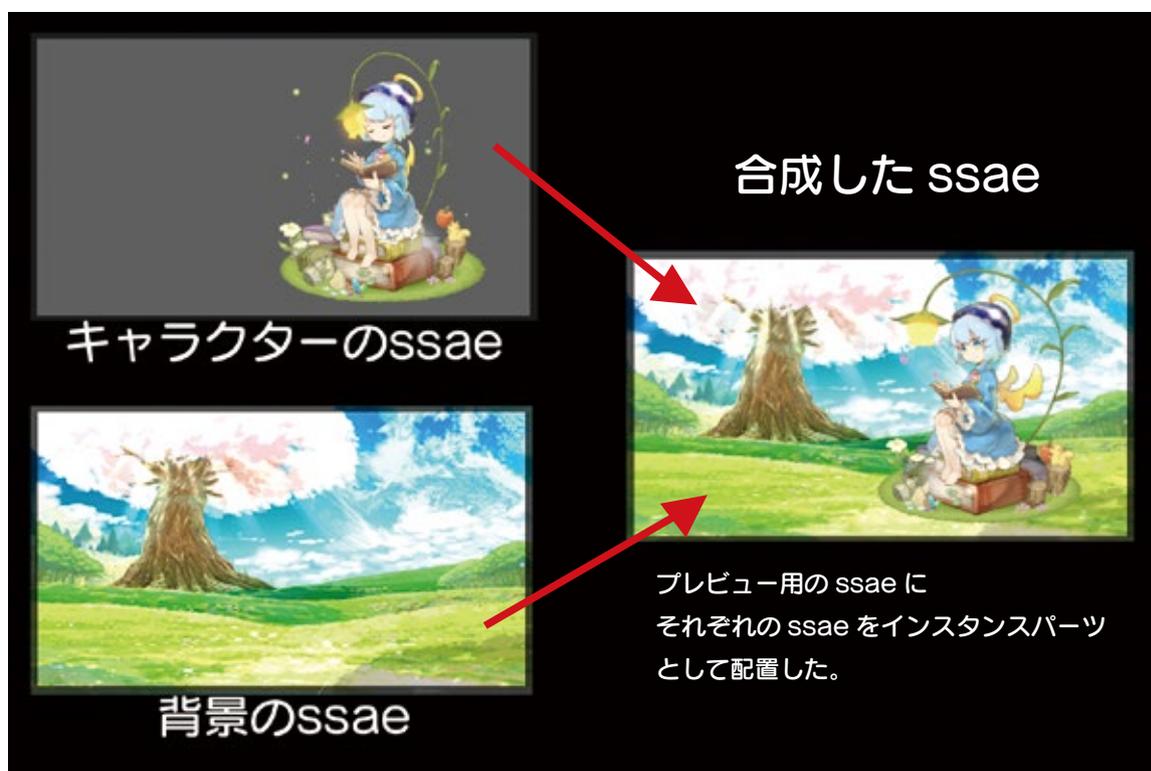
## インスタンス機能で複数のアニメを合成しよう

2人のキャラの合成技など、パーツが2倍になってしまい、管理が大変です。そんなときはインスタンス機能を使ってアニメーションを合成してみましょう。

## インスタンスパーツって何? インスタンス機能の基礎知識

インスタンスパーツは他のアニメーションデータを参照するパーツです。

たとえば、キャラクターと背景のアニメーションを別々の ssae で作成しておき、別途作成しておいたプレビュー用の ssae に、それぞれをインスタンスパーツとして配置、合成して、ひとつのアニメーションを作成することができます。



インスタンスパーツには、以下のような特徴があります。

- 通常パーツと同じように配置可能
- 「位置調整」「回転」「スケール」も設定可能
- データサイズを小さくできる
- パーツサイズはソースアニメーションの基準枠サイズ
- 独自にアニメーション再生可能

## インスタンス機能のメリット

インスタンス機能を活用することで、以下のようなメリットがあります。

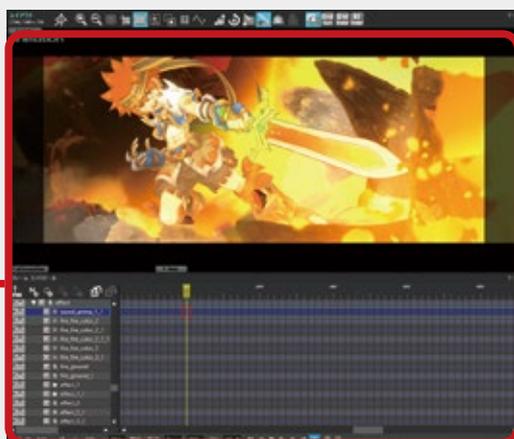
- 簡単に複数バリエーションが作れる
- 独立再生が行える
- シーン構築が簡単にできる
- シーケンスの組み合わせが行える

## HINT インスタンスパーツの活用事例

### より迫力のある演出に

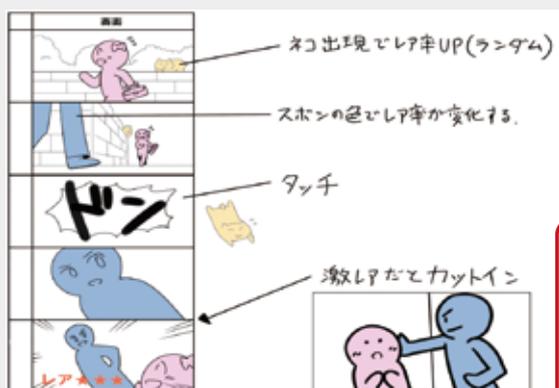
インスタンスパーツはスケールを変更することも可能です。一つ汎用性のあるアニメーションを作成すると様々な演出強化に使用できます。

炎のアニメーションを  
別途作り、インスタンスパーツとして  
配置しました



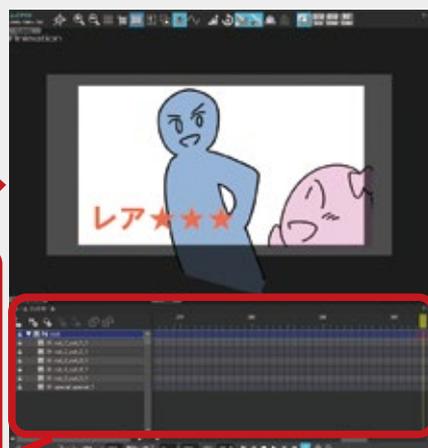
### シーンをつなぎ合わせて1本のムービーに

ワンカットずつ ssae を作成し、インスタンスパーツでつなぎ合わせてムービーを作成することができます。1シーンごとに管理ができるため、修正、調整が簡単です。ガチャ演出など、条件によって分岐する演出を作成するときにもお勧めです。



ワンカットごとにアニメーションを作成し、インスタンスパーツで配置。

つなげて一本のムービーを作成しました



# インスタンスパーツをアニメーションに組み込む

作成したアニメーションデータをインスタンスパーツとして組み込みます。

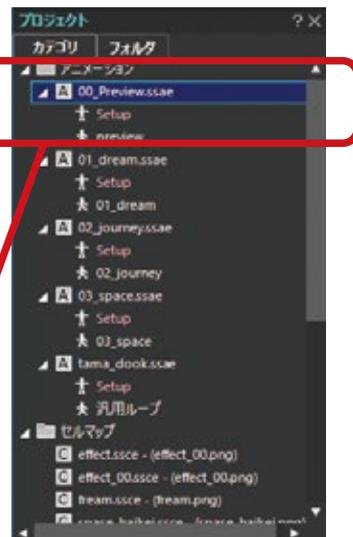
合成用に新しく ssae を作成します。

今回はプレビュー用の ssae を別に作成します。

これは作成したアニメーションを確認するためのファイルになります。

別のファイルにすることで、パーツ構成がシンプルになり、修正や確認が容易になります。

作成したアニメーションにインスタンスパーツを直接組み込んで問題ありません。

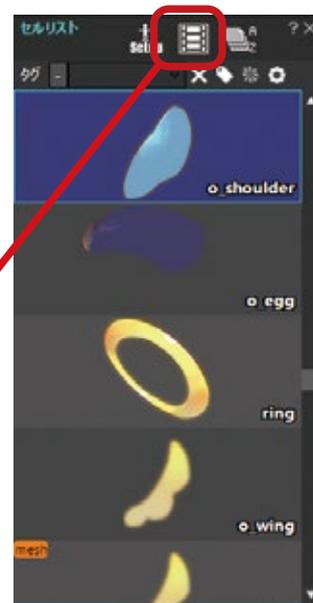


1 合成用の ssae を別に新規作成

新たに作った ssae を選択し、

セルリストウィンドウのリスト切り替えボタンをクリックして「アニメリスト」に切り替えます。

アニメリストに、現在選択している ssae 以外のアニメーションデータが表示されます。



2 リスト切り替えボタンをクリック

3 セルリストが「アニメリスト」に切り替わりました



セルを配置するのと同じように「アニメリスト」からドラッグして配置します。

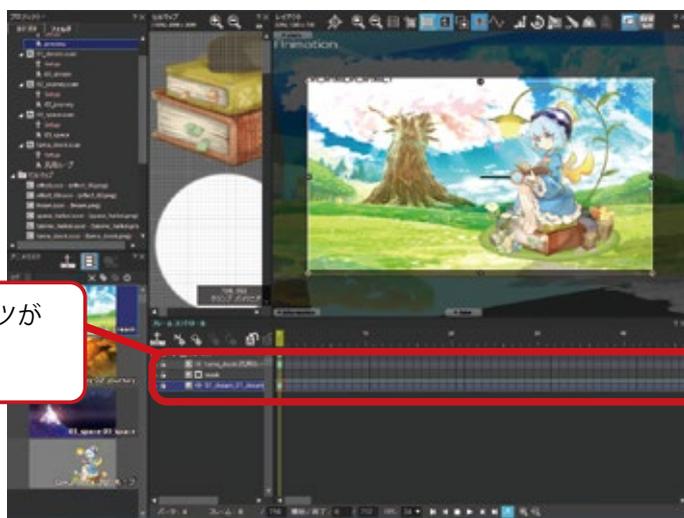
右図ではフレームコントロールウィンドウにドラッグしていますが、レイアウトウィンドウ上にドラッグすることでも配置可能です。

- 4 アニメリストから  
インスタンスパーツとして組み込む  
アニメーションをドラッグ



これでインスタンスパーツの組み込みは完了です。  
再生するとそれぞれのインスタンスパーツが  
1つのアニメーションとして再生されます。

- 5 インスタンスパーツが  
追加されました



## HINT インスタンスパーツに使えるアトリビュート

インスタンスパーツは通常のパーツ同様、フレームコントロール上で制御することができます。  
ただし使用できるアトリビュートは以下に限られます。

- X/Y 座標
- X/Y/Z 回転
- X/Y スケール
- 不透明度
- 優先度
- 非表示

## インスタンスアトリビュートを使ってみよう

インスタンスパーツ独自の「インスタンス」アトリビュートが存在します。  
このアトリビュートを設定することで、インスタンスパーツの再生制御を行うことができます。

- 再生区間
- 再生速度
- ループ設定
- 再生方向

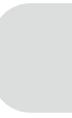
を指定できます。

例えば自動車のタイヤを回ったままにしたい場合は、ループ設定の「独立動作」と「無限ループ」にチェックを入れることで、タイヤのインスタンスパーツは組み込まれたアニメーションを無視し、ソースアニメーションにもとづいて、無限ループで再生されます。



インスタンスパーツの活用例については  
ヘルプセンターもご覧ください。  
[http://www.webtech.co.jp/help/ja/spritestudio/guide/tips/instance\\_example/](http://www.webtech.co.jp/help/ja/spritestudio/guide/tips/instance_example/)





## EX-1 PSDtoSS を使ってみよう

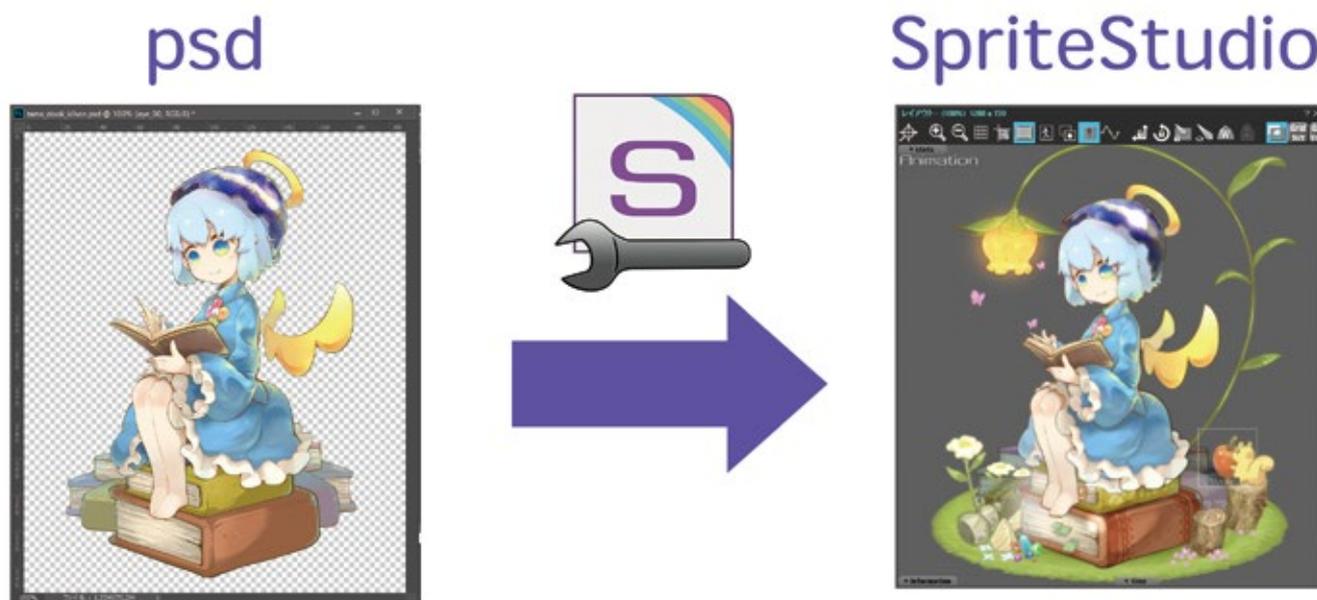
イラストを準備して、パーツ分けをして、テクスチャを作って、レイアウトして……。アニメーションさせるための準備は手間がかかりますが、PSDtoSS を使えば大幅に作業を削減、効率よく準備できます。

### 心強い味方 PSDtoSS とは?

「PSDtoSS」は SpriteStudio の公式補助ツールです。Photoshop で作成した「PSD ファイル」を、SpriteStudio の各種ファイルへコンバートすることができます。

PSD のレイヤー情報をもとにセル・セルマップを作成し、最適なサイズのテクスチャを作成。

PSD で作成したレイアウトそのままに、アニメーションデータにパーツ配置まで行います。すぐにアニメーション編集に着手することができます。



PSD を変換して SpriteStudio の各種ファイル形式にコンバートしてくれます。

PSDtoSS を使うことでかなりの手順を省略し、すぐにアニメーションを作成する作業に移ることができます。無料で利用できるなので、ぜひ活用してください。

PSDtoSS は SpriteStudio Ver.6 に対応した「PSDtoSS6」が最新版となります。

本誌では「PSDtoSS6」をもとに解説を行っています。

PSDtoSS6 はヘルプセンターからダウンロードすることができます。

<http://www.webtech.co.jp/help/ja/spritestudio/download/for designer/>



1 イラストを作成します



2 パーツをレイヤーごとに分けます  
このレイヤーがそのまま  
セルやパーツとして登録されます

3 PSDtoSS を起動します

4 PSD をドラッグ・ドロップします



5 出力フォルダを選択します

6 ssae 出力、spsj 出力に  
チェックを入れます

7 「コンバート開始」を押して  
エラーが出なければ完了です

## EX-2 SsKeyBindingEditor でショートカットを設定しよう

ここまで読めばあなたもすっかり SpriteStudio を使いこなせるようになったはず。そんな中、ちょっとかゆいところに手が届かない……と感じることも? そんな時は自由にショートカットを設定して、作業スピードをどんどん上げていきましょう!

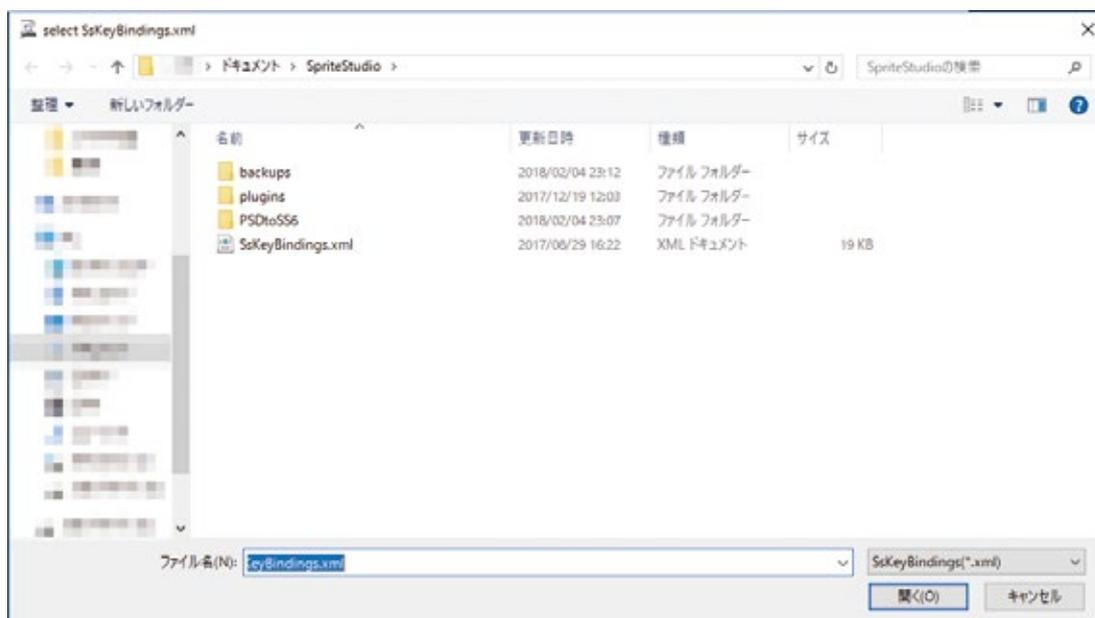
SsKeyBindingEditor は SpriteStudio に対応した、ショートカットキーカスタマイズツールです。搭載されているショートカットキーの確認と、カスタマイズを行えます。どのようなショートカットキーが採用されているか把握したり、自分の使いやすいようにカスタマイズすることで、作業効率の向上が見込めます。

SpriteStudio ver.6 に対応しているのは「SsKeyBindingEditorV6」です。本誌では「SsKeyBindingEditorV6」をもとに解説を行っています。SsKeyBindingEditorV6 はヘルプセンターからダウンロードすることができます。  
<http://www.webtech.co.jp/help/ja/spritestudio/guide/tool/sskeybindingeditorv6/>



## SpriteStudio を起動し「SsKeyBindings.xml」を作成しよう

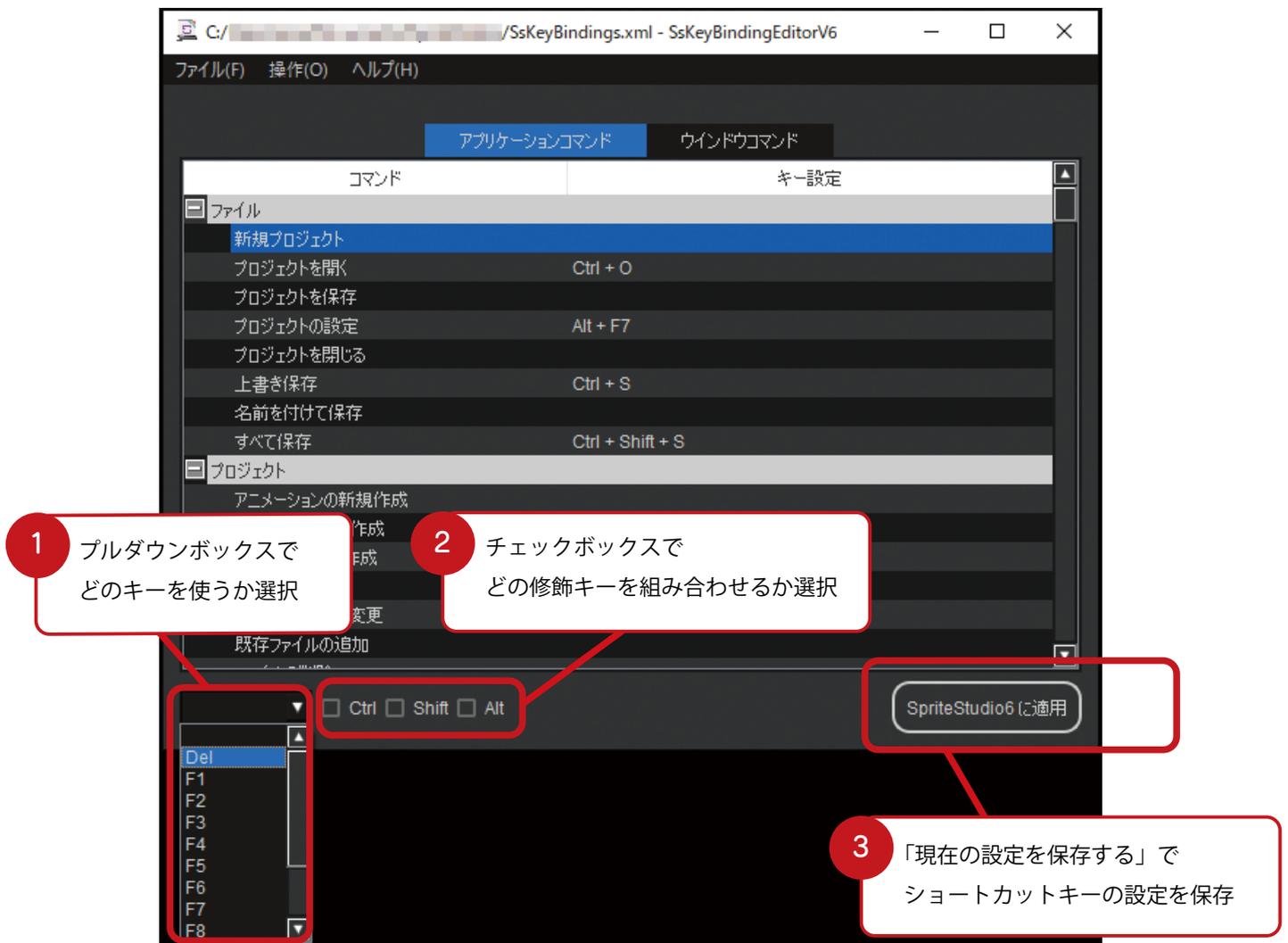
「SsKeyBindings.xml」は SpriteStudio のキー設定ファイルです。SpriteStudio はこの「SsKeyBindings.xml」を読み込みます。別のキー設定ファイルを読み込むように指定することはできません。「SsKeyBindings.xml」は、SpriteStudio を起動すると自動的に作成されます。



## ショートカットキーの編集を行おう

SsKeyBindingEditor フォルダ内にある「SsKeyBindingEditor.exe」をダブルクリックすると「SsKeyBindingEditor」が開きます。

各操作に対して設定されているショートカットキーの一覧が表示されます。

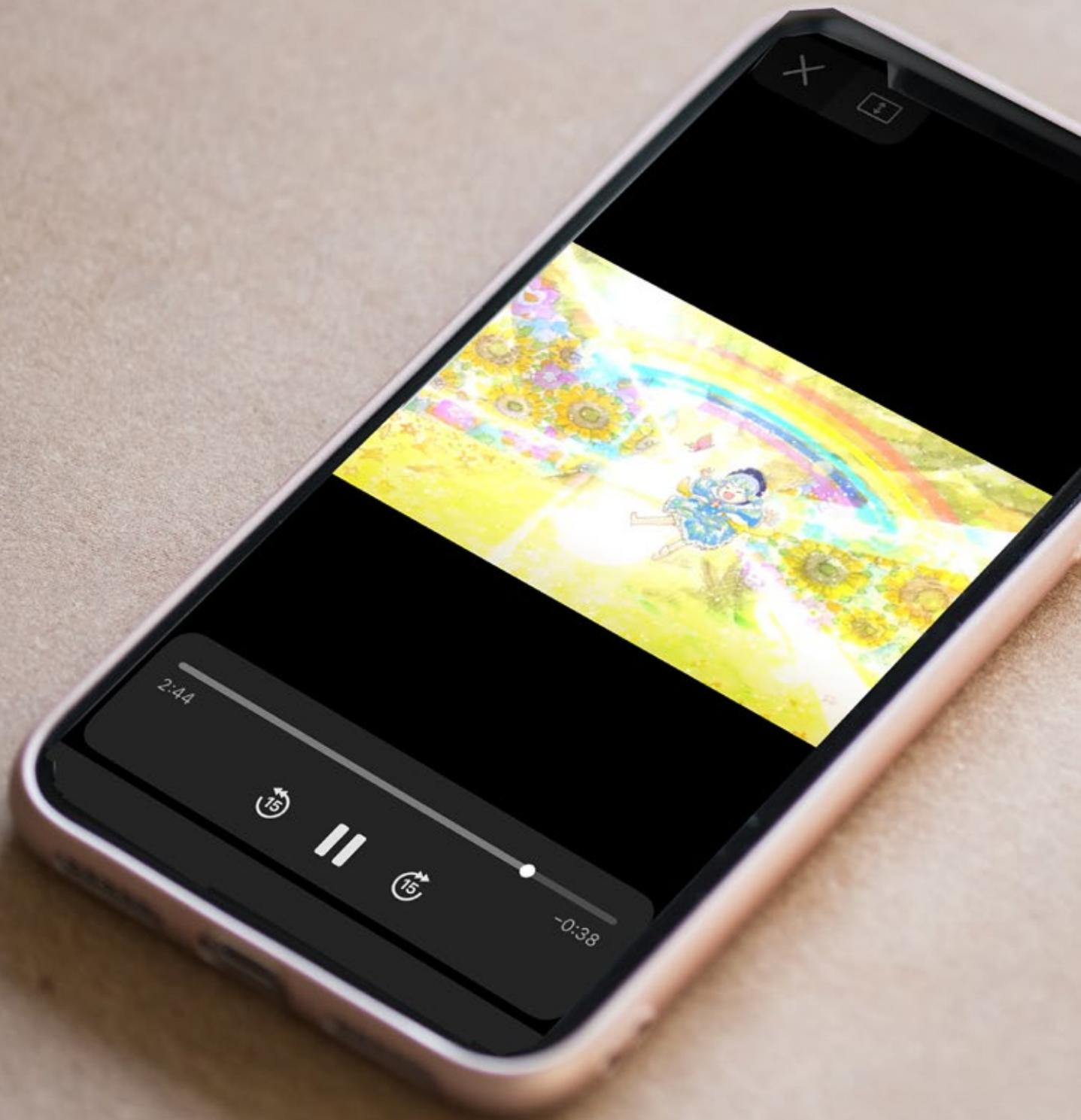


ショートカットキーの設定が完了したら「SpriteStudio6 に適用」ボタンをクリックし、SsKeyBindingEditorV6 で設定したショートカットキーを「SsKeyBindings.xml」に保存します。

コマンドで設定しているショートカットキーが重複するなど、設定できないキーがある場合、エラーダイアログが表示されます。



# SpriteStudio の活用事例



池田陽朗のアニメーション作成メイキング  
～『てのびらの宇宙』はどのようにして作られたか

	2015	2016	%
	812		
	918	108,287	
	939	91,938	
	128	125,819	
	278		
	221		
			+31%
			+42%
			+18%
			+18%

# 池田陽朗のアニメーション作成メイキング 『てのひらの宇宙』は どのようにして 作られたか

SpriteStudio といえばゲーム開発の現場で使用されることが主でした。

しかし SpriteStudio エヴァンジェリスト池田陽朗氏により、SpriteStudio の多彩な機能と表現力はアニメーション作品にも活用できることが証明されました。

今回は第 27 回 CG アニメコンテストにて入選した 3 分強のアニメーション作品『てのひらの宇宙』のメイキングをご紹介します。

手書きや実写を取り込んだ、良い意味で「SpriteStudio で作られたとは思えない」短編アニメーションの制作の舞台裏をご覧ください。



## 絵コンテを作る ～絵コンテは全体の設計図～

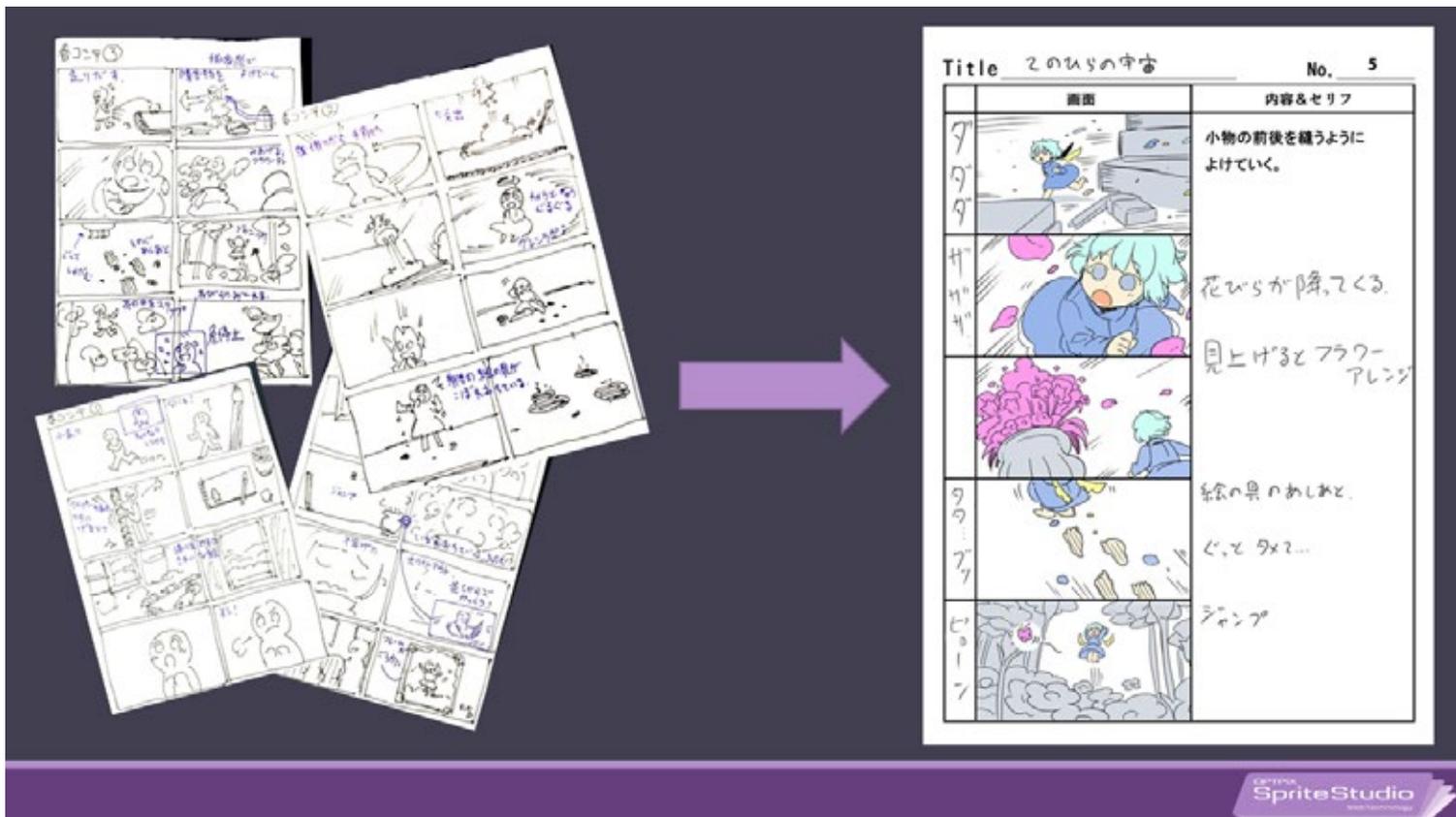
まずはじめに、作品のアイデアを「絵コンテ」に描き起こします。

絵コンテを用意することで、作品の完成形をイメージでき必要な素材や演出が分かります。複数の人と共同で作る場合には、作品のイメージを共有することにも役立ちます。

プロットやアイデアを整理して絵コンテにします

描きながら、浮かんできて良かったアイデアは次々に取り入れます。

今回の作品では、クロッキー帳から水彩絵の具で直接描いたキャラクターが飛び出す演出をSpriteStudioで動かそう、フラワーアレンジの中を駆け回らせてみようといったアイデアを盛り込んでいきました。



絵コンテで全体のイメージを固めていく。このコンテがアニメーション制作における設計図となっていく。

## 絵コンテから必要な絵を割り出す

絵コンテが完成したら、必要な素材を洗い出します。

キャラクターを動かす場合、絵は多いほど表現の幅が増えるので良いのですが今回はコンテストに応募するための制作期間が2ヶ月と短期なので、表現に必要な分だけの絵を描き起こすことにしました。

絵コンテを並べながら、各カットの絵作りが必要であると想定する部分に印をつけておきます。(画像のオレンジ色の部分が素材の必要な場所)

キャラクターの絵が必要な部分に印をつけたら、実際に絵を描き起こしていきます。

## 原画を 割り出す

- 描き起こし部分に印
- パーツの流用を考慮

Title 乙のいのちの宇宙		No. 1
画面	内容&セリフ	
	たまごに SSのアイコン、はいちゅうちゅうがとまる。	
パリン 	たまごから元気が現れる キャラ	
チキチキ... 	ちゅうをみいかけ。	
	実写の筆を横切る	
ドカット 	よそ見でフチに激突	

絵コンテから必要な原画(素材)を割り出す様子。絵コンテが全体の設計図として重要なのがよくわかる。

## 絵コンテを作成するメリット

- 作品の全体像・作業量の把握ができる → 作業の効率化
- ほかの人とイメージの共有ができる → 相談、作業分担が容易に
- 必要な素材の割り出し・修正回数の減少 → 不要な作業の削減
- 形にすることで実現可能か検討可能
- さらにアイデアを練りこむなどクオリティ向上に役立つ
- 資料として残るため、完成作品と比較して次回に活かすことが可能

## 原画の描き起こし

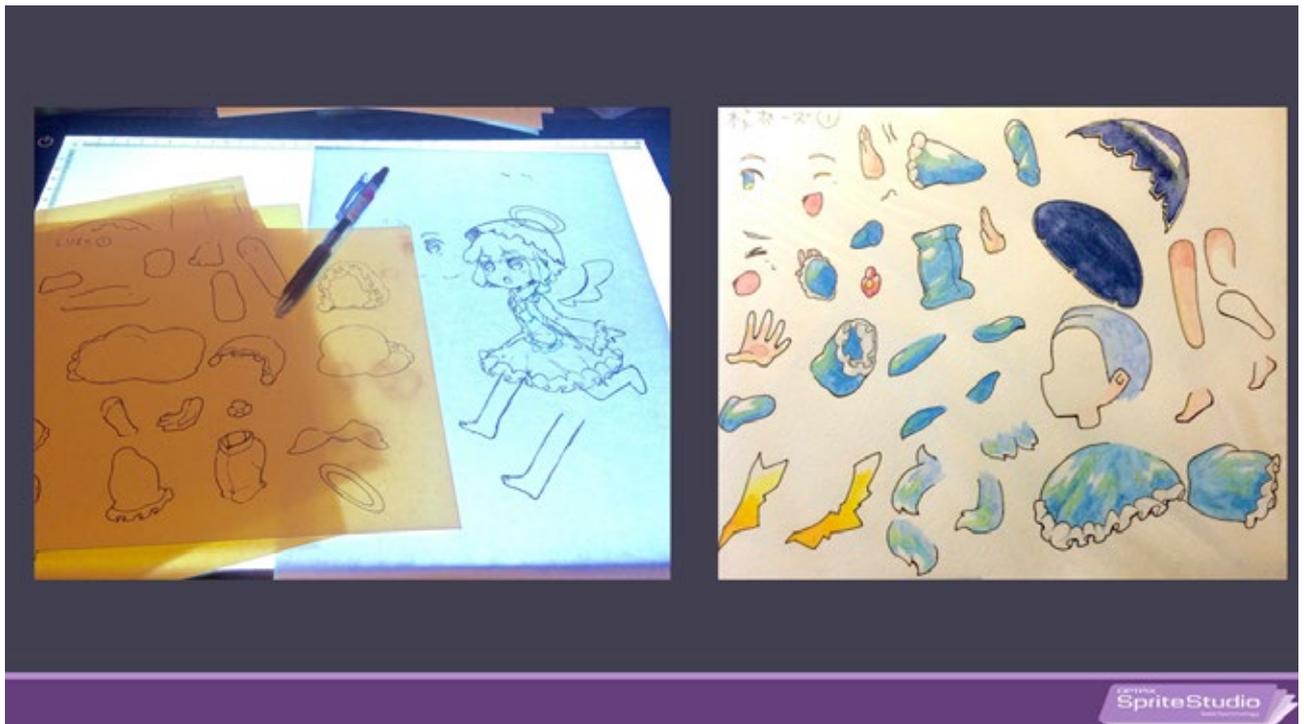
絵コンテをもとに必要な絵を描いていきます。まずはキャラクターラフを作成し、どのように動かすかなど、パーツの構成を検討しながら作成していきます。

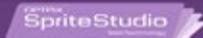


ラフの様子。キャラクターデザインを練りつつパーツ分けや動きを意識したラフになっている。 

ラフを描いたら、トレース台を使って別紙に清書します。  
水彩で色を塗ることを考慮して、油性ボールペンを使用しました。  
この時、SpriteStudioで動かす部分（顔、手、足など）を分けながら描きます。

清書が完了したら、色を塗って完成です。



今作ではアナログ作画だったが、もちろんデジタル作画でも同様の方法でパーツ分けが可能 

## パーツ分割をする～アニメーションを想定したパーツ分割～

原画の描き方は、アニメーションのパーツ数やデータの構成に大きく影響を与えます。

あらかじめ構成がシンプルになるようにデザインすることで、

操作するデータを抑え作業時間を短縮することができます。

例えば袖の中の腕まで動かそうとすると、フリルや服で手をサンドイッチさせる必要がありますので、画像上段のような、6分割構成になります。

この構成だとフリルもアニメーションすることが可能なので、

非常に豪華なアニメーションを作成することが可能です。

しかし、すべての部位を可能なかぎりパーツ分けをすると、データ量が増え、

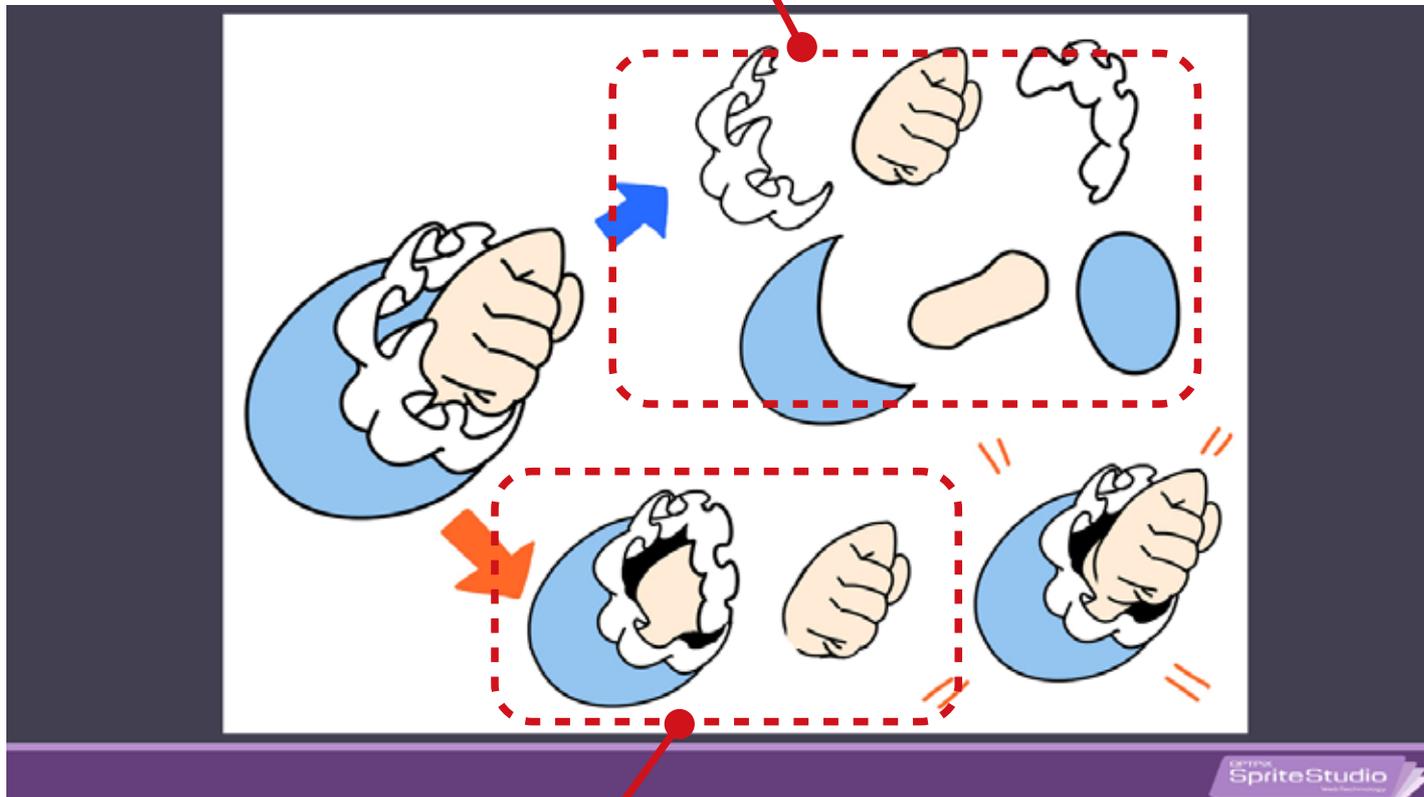
パーツをリストから探す手間がかかるなど、作業量・修正作業がどんどん増えていきます。

パーツごとにアニメーションの目的（やりたいこと、見せたいこと）を意識しながら、

パーツ分割数を決めていきます。

今回は手の動きは重要ではないと判断し、下段のようにパーツ数を2つに絞りました。

袖の中・フリルまで動かそうとすると  
パーツは6分割となる



今回は手のアニメーションを重視して  
いないため、最低限に絞った

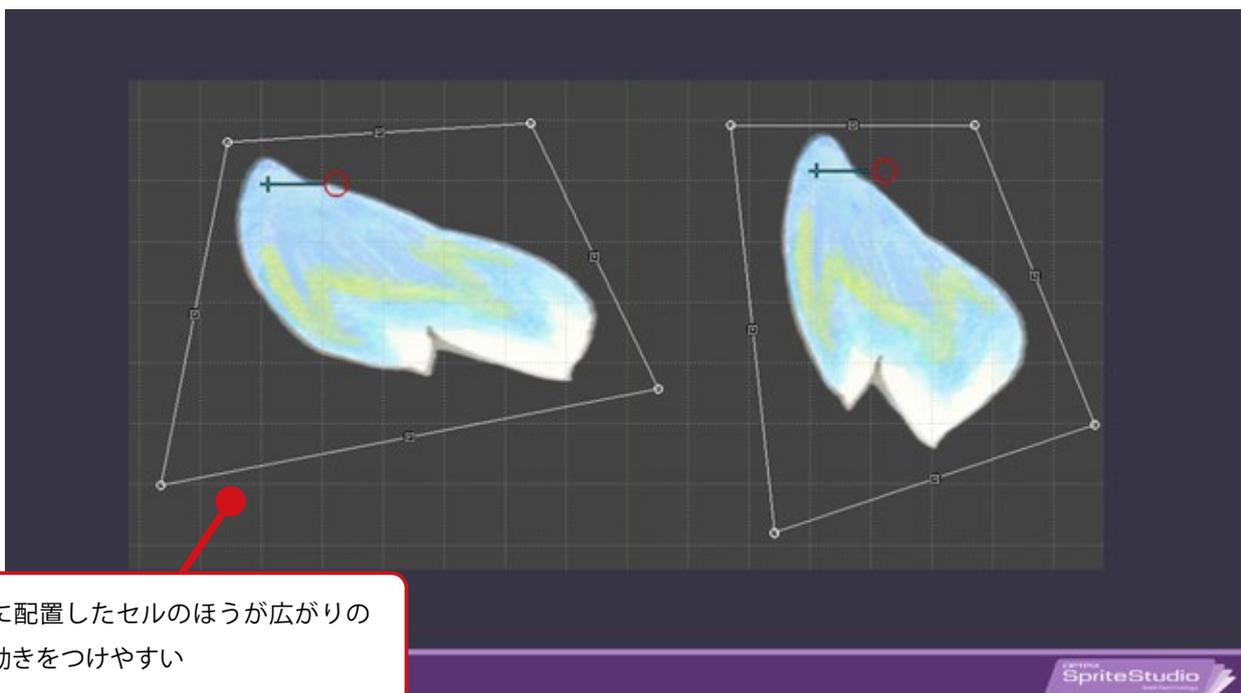
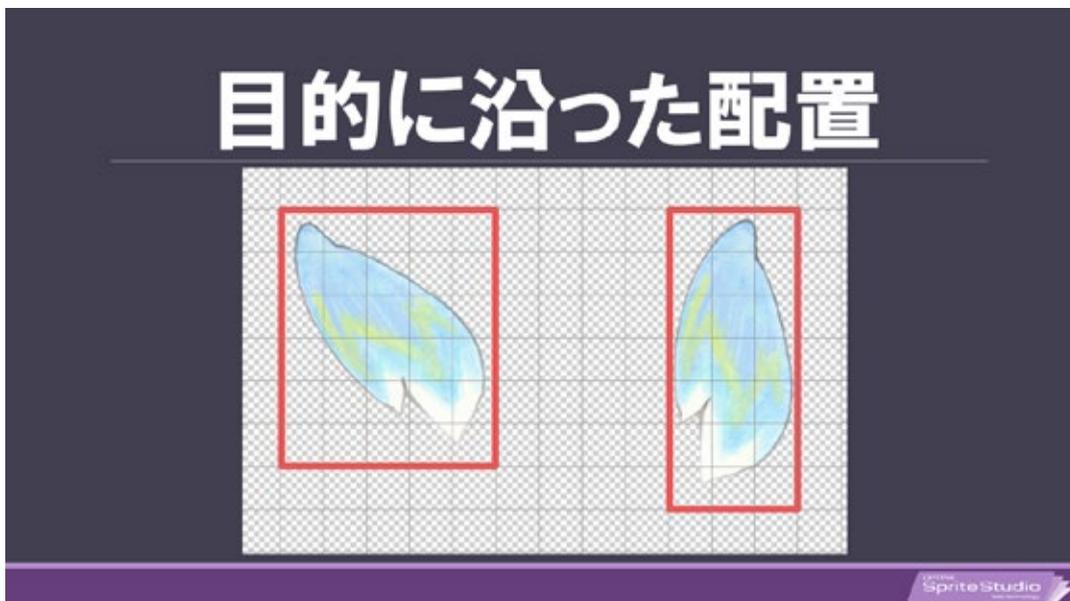
## セルマップ作りのコツ～目的に応じた配置をする～

各パーツをセルマップへ配置にする際に、少し工夫しているところがあります。  
セルの『アニメーションの目的に沿った配置について』です。

水平に配置されたパーツとナナメに配置されたパーツとでは SpriteStudio での自由変形で表現できることが異なってきます。  
そこで、「揺れモノはナナメ配置」をしています。

髪や布といった、パーツの根元と先端が時間差でズレて揺れ動くようなものを 1 枚絵で表現したい場合は、そのポイントが四角形の対角線あたりに配置するようにします。  
こうすることで、パーツの先端部分を曲げやすくなります。

画像は髪の毛のパーツの様子です。  
斜め配置にするほうが先端が有機的に曲がっていることがわかります。



アニメを生かす背景作り

## 背景の情報量を操作する

「てのひらの宇宙」の背景はクロッキー帳やフラワーアレンジなどといった小物を机の上に並べて舞台を作り、カメラで撮影したものを使用しています。

ですが、写真そのままを使用せずに、デジタルで加工を行いました。

時計の持つ情報である、「時間」というワードには特別な意味と強さがあると自分は考えています。そのような、見る側に影響を与え、物語に関係のないモチーフは消去します。

アニメを生かす背景作り

## 重要なモチーフは目立たせる

鉢の大きさと花のボリュームが違います。シーンを見た時にすぐ大事なモチーフは何なのかを知らせるためです。主人公の小人感を出すためにアオリアングルで撮影し、フレーミングは上辺ぎりぎりまで大きくします。

花の形を整えることもそうですが、配置の工夫として右側の紫色の花よりも映る面積と高さを上げるようにして対比も作ります。

壁や天井の情報をそぎ落として  
すっきりさせている

### 不要な情報は描かない



加工前



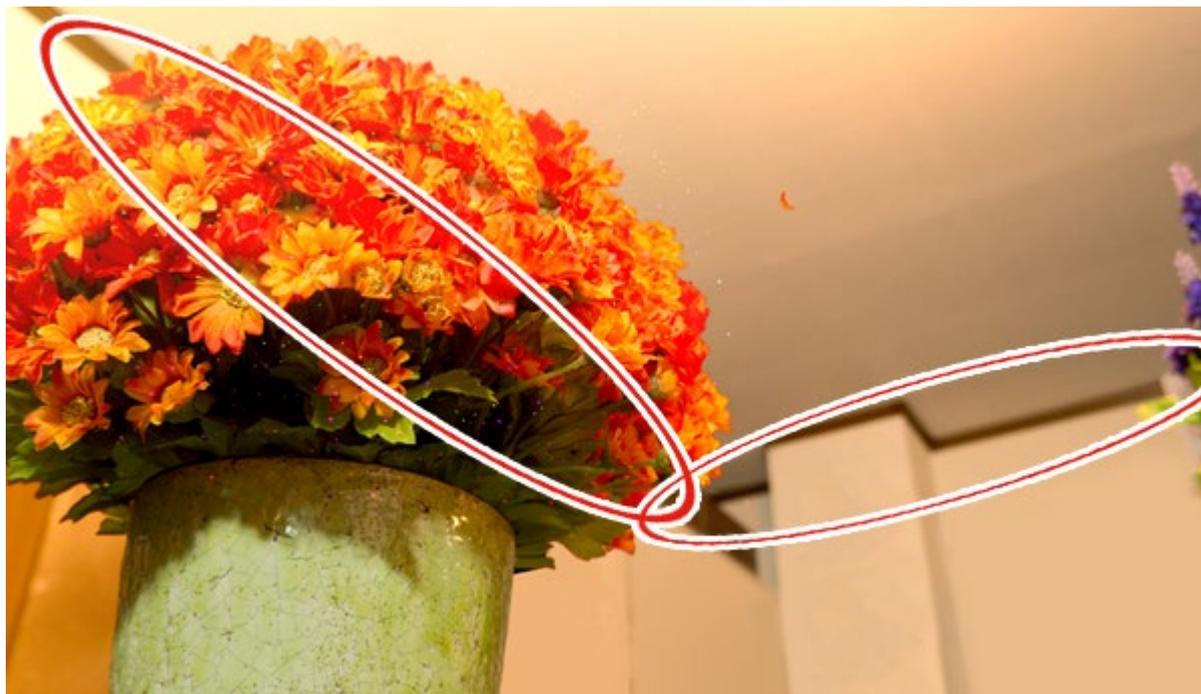
加工後

花のボリュームが増えている

## アニメを生かす背景づくり 「直線」を利用する

絵の情報では直線は目線の流れを促し、とても強い影響力を持ちます。  
上図の場合では天井の茶色の縁が空間の奥行きや方向性を助けています。

さらに線と線が交わる交点に視線が集約しますので  
物語のキーとなるもの（この場合は花）を置いておくのも有効です。



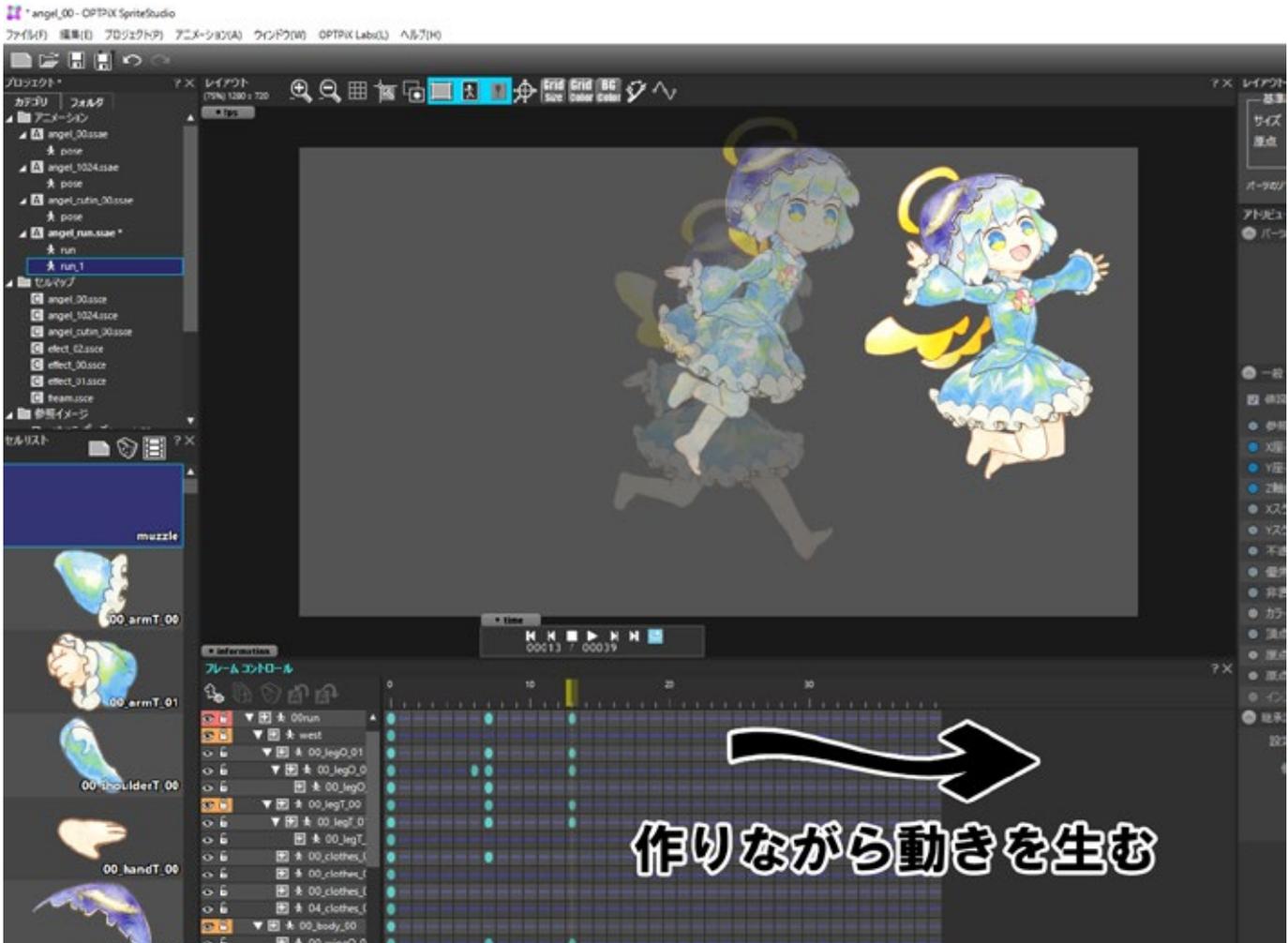
天井の茶色の縁が奥行を出し、キャラクターの動く方向を示唆してくれる。  
背景の情報をそぎ落としているおかげで、より一層「直線」を意識しやすくなっているのがわかる。

## 魅力的なアニメーションに見せるコツ ふたつのアニメーション手法

キャラクターを動かしていく上で、大きくふたつの方式に分けることができます。  
これらの手法を目的に応じて使い分けたり、時には組み合わせてアニメーションを作ります。  
今回は「ポーズ・トゥ・ポーズ方式」で全体を仕上げた後、「ストレート・アヘッド方式」で細部の調整を行いました。

### ストレート・アヘッド方式

とにかく動かし始めて、その都度起きるキャラクターの動きをたどっていく方式。  
自然な流れに沿ってアニメーションをつけていきます。



### メリット

のびのびとクリエイティブに作ることができる。  
作りながら無意識にアドリブの動きが入ったり、生命感が宿りやすい。

### デメリット

行動をキャラクターの意思に委ねるため、先のことを把握しづらい。  
制作時間と修正コストがかかる。

## ポーズ・トゥ・ポーズ方式

重要なポーズをあらかじめ先に作り、そのポーズの間に中割りを作っていく方式



### メリット

計画的に、比較的スムーズに制作が進められる。修正しやすい。

### デメリット

計算ずくで機械的な動きになりすぎる傾向がある。

ふたつの方式にはそれぞれ相反するメリットとデメリットがある。

SpriteStudioは「ポーズ・トゥ・ポーズ方式」を得意としているが、それだけではあからさまに機械的な動きになってしまうため、キーを増やすことを面倒がらずに「ストレート・アヘッド方式」も用いて作りこむことで、より生き生きとしたアニメーションを作成することができる。

## 2D アニメーションに『空間』を作り出す ～空間に奥行きを出す工夫～

2D アニメーションを作っていると無意識にY軸（上下）とX軸（左右）の動きに縛られたものを作ってしまう、平面感が強調されがちです。画面に「奥行き空間」を作るためにはキャラクター、背景、エフェクト、カメラなどをZ軸（手前や奥）に操作しなければなりません。

では実際にどのように動かすことで空間に立体感を出すのか？キャラクターがフラワーアレンジの中を動き回るシーンを例にポイントを解説します。

### 舞台に階層を作り、奥行きを出す

キャラクターを中景とし、背景に『近景』と『遠景』を階層的に用意することで絵に奥行きが生まれます。

もっと奥行きを出したい場合には、建物や物の配置で直線的にパースを持たせたり、空気遠近法やカメラの絞りによるぼかし、スクロールスピードに近景と遠景で変化をつけるなどの手法があります。



キャラクターを挟むように近景と遠景を作成。3つの階層により世界に奥行きが生まれている。

## 奥～手前に移動する時は「曲線」で移動する

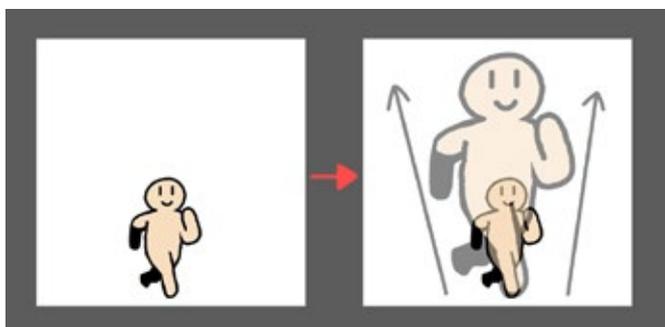
キャラクターが奥側から手前側へ移動してくるときにはX、Y、Z軸の動きを混ぜて行くと、より舞台の空間を広く見せることができます。



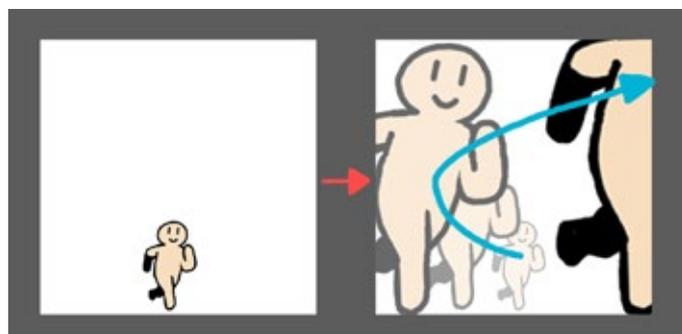
X、Y、Z軸の動きを混ぜずに、拡大のみでもキャラクターが近づいてくる、遠のくという表現を作ることができます。

しかし、奥側から手前側に走ってきているようにも見えますが、「キャラクターが巨大化している」ようにも見えます。

これは「キャラクター自体の座標変化の情報量」よりも「キャラクターが拡大する情報量」の方が勝ってしまっているために起きてしまう現象です。



拡大のみの表現では「移動」ではなく「大きさの変化」にしか見えない可能性がある。



拡大のみではなく、実際に座標も動かして表現することで、空間に奥行きも生まれ、誤解も生まれません。

## フレームイン・フレームアウトを使う

舞台の広がりを感じさせるためには、「画面に写っている外側にも世界は続いている」ということを伝えなければなりません。

そのためには様々な方法があるのですが、その中のひとつとして「フレーム枠を使う」方法があります。

あえてキャラクターを一度画面外へフレームアウトさせます。

その後、アングルが変わりながらキャラクターが画面内へフレームインすることで、直接キャラクターが身を翻すアニメーションを画面内で作らなくても、その間が補間されているように感じさせることができます。

さらにこのシーンでも、先程の「曲線の動き」でキャラクターを移動させることで、空間作りを補助しています。

キャラクターがフレームイン・アウトを行うことで、「フレームの外にも世界がある」ように感じさせることができる



## まとめ ～制作を終えて～

制作前はこれまで SpriteStudio で 1 ～ 10 秒ほどのアニメーションのみを作ってきて、いきなり 3 分ほどの物語を内在したアニメーションが作れるのかどうか不安でした。結果的に形にはなったものの、「伝える」ことの難しさを痛感しました。

絵をアニメーションさせ見る側の感情を呼び起こせるような物語をいつか作れるようになりたいです。

最後まで読んでくださりありがとうございます。

OPTiX SpriteStudio は皆さまのさまざまなアイデア、クリエイティビティに応える素敵なツールです。

ぜひ、一度チャレンジしてみてください。

## 参考文献

アニメーターズ・サバイバルキット（グラフィック社）  
映像の原則（キネマ旬報社）  
filmmaker's eye（ボーンデジタル）

●本記事は OPTiX Labs Blog 「ゲーム以外での OPTiX SpriteStudio の活用事例～「てのひらの宇宙」メイキング～」の再編集版です。  
<http://www.webtech.co.jp/blog/products/spritestudio/9952/>



●ヘルプセンターでは本作品も含む、動画作品の一覧を公開しています。  
本誌を片手に「どこでどんな機能を使っているのかな？」と想像しながらご覧いただくと、SpriteStudio でのアニメーション制作方法への理解が深まること請け合いです。  
ぜひご覧ください。

[http://www.webtech.co.jp/help/ja/spritestudio/movie/#ss\\_Product](http://www.webtech.co.jp/help/ja/spritestudio/movie/#ss_Product)



# OPTPiX SpriteStudio BEGINNERS 2

だれでも作れる！2Dアニメーションの基本のき

2018年4月30日 初版 第一刷発行

2019年9月5日 pdf版発行

著者 株式会社 ウェブテクノロジー

監修 池田 陽朗 (透き通る風: transparent-wind.com)

編集 瀬良 茜・株式会社 ウェブテクノロジー SpriteStudio 開発チーム

発行人 小高 輝真

発行所 株式会社 ウェブテクノロジー

150-002 東京都渋谷区渋谷 1-7-7 住友不動産青山通ビル 9F

<https://www.webtech.co.jp>

© 2019 Web Technology Corp.

夢を描いて、動かそう。



[www.webtech.co.jp/spritestudio](http://www.webtech.co.jp/spritestudio)