



Pinball Construction Kit

November 2015

Joe Conley & Rob Gray
Epic Games Japan

概要

ピンボール コンストラクション キット (PCK) は、日本で行ったUnreal Fest 2015のプレゼンのために制作されたサンプルです。単なるピンボールゲームを作るというよりも、UE4のブループリントとエディターのカスタマイズで、複数のピンボール台が作れるフレームワークを作るのが目的でした。

このプロジェクトを参考にして UE4 の勉強をしていただけると嬉しいし、コミュニティのみなさんがどんなに面白いピンボール台を作るかをとても楽しみにしています。何か作ったら是非フォーラムでみんなとシェアしてください！

はじめに

1. StarterTableというマップファイルをベースにすると、新しいテーブルをすぐに作ることができます。
2. ボールをテーブルに打ち出す為に、BP_Plunger をテーブルに必ず配置する必要があります。そして、落ちたボールを反応するBP_Drainも必要です。
3. そして、ターゲット、バンパー、ライトなどを配置して楽しんでください！

ピンボール台を作ろう

パーツを簡単にドラッグ&ドロップして、すぐにピンボール台が作れるようなシステムを作りました。このプロジェクトには複数のカスタマイズ可能なパーツがあるので、コードなどを触らなくても配置してパラメータを設定するだけでピンボール台が作れます。

しかし、更にブループリントを拡張して機能を追加したい人もきっといると思うので、プロジェクトの構成を説明します。

MyPinballGameMode

GameMode がゲームのルールやゲームプレイの機能をコントロールします。基本のシステムが既にできていますが、是非それを拡張して特別なルールなどを追加してみてください。デフォルトの MyPinballgameMode は以下を行います：

- ゲーム開始時に、マップに配置されたBP_Plungerのオブジェクトを探して変数に保存します。これはボールをシュートするために使用されます。
- 同様に、マップに配置された BP_Indicator_Special のオブジェクトを探して変数に保存します。BP_Indicator_Specialはゲームが特別モードに入るときにライトアップされるライトです。特別モードの詳しい説明については、以下のBP_Indicator_Specialのセクションをご覧ください。
- プレーヤーのスコアなどを表示する UMG の HUD をビューポートに追加します。
- 台のスタートアップで、デフォルトの開始音（音楽）を再生し、台にあるすべてのライトを順番にチカチカさせます。これが終わったら最初のボールがロードされます。
- デフォルトで、ゲームが 30 秒間の Ball Save モードで開始します。このモードがアクティブのときに、ボールがリターンホールに落ちても、自動的にプレイエリアに戻されて、ロストボールとしてカウントされません。

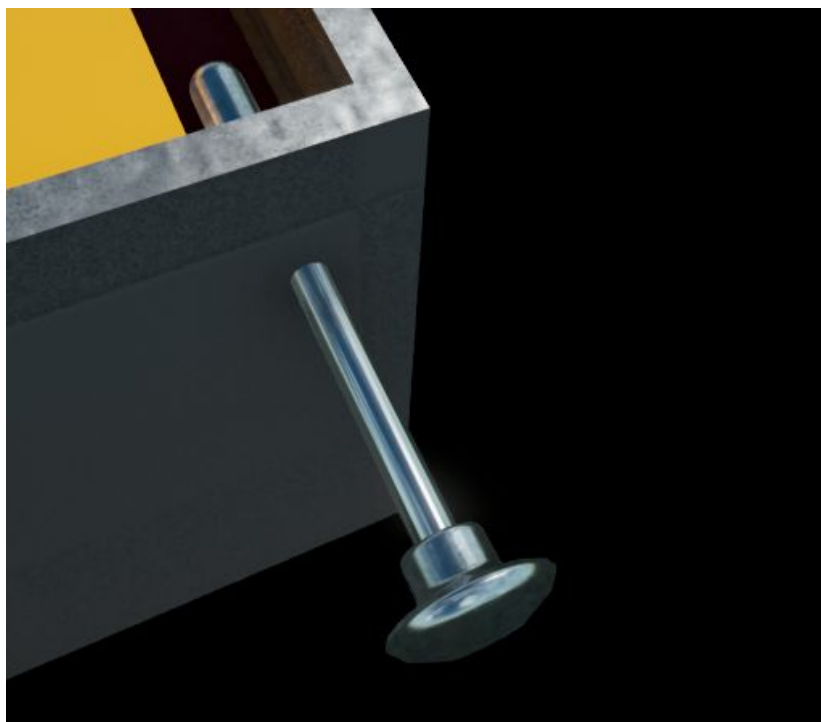
PinballPlayerController

デフォルトの PinballPlayerController は台に配置されたすべての BP_Flipper のオブジェクトを探して変数に保存します。そして、左フリッパーか右フリッパーの設定に対して、それぞれ左と右の SHIFT キーで操作できるようにアサインします。そして、現実のピンボール台の場合、ボールの動きを操作するために台を叩いて揺らす技があります。この叩くアクションは PlayerController で左と右の CTRL キーで処理します。しかし、プレイヤーが台を叩き過ぎると、ペナルティで Tilt モードに入り、フリッパーの操作ができなくなってしまいます。

ピンボール台のパーツ

PCK にはブループリントで作られたそのまま使えるカスタマイズ可能なピンボールマシンの機能があります。自分のテーブルにドラッグ&ドロップして配置して、設定をお好きなように変更して、簡単に素早く自分のピンボールテーブルを作ってみてください！

BP_Plunger



これは台にボールをシュートする機能です。すべての台に配置する必要があります。

BP_AutoPlunger

これは通常、ピンボール台の右と左のガターにあります。ボールがガターに落ちても、これは自動的にボールを打ち出して、ボールをプレイエリアに戻します。

- **NumberOfSaves** - プランジャーがボールをセーブするためにアクティベートする回数

BP_Drain

落ちたボールを反応するボールのリターンです。すべてピンボール台に配置する必要があります。フリッパーの後ろに配置するべきです。

BP_Flipper



ボールを跳ね返すフリッパーです。ゲーム開始の際に、これが自動的に PlayerController に登録され、左と右のSHIFTキーで操作ができます。

- **Start Rotation** - 最初の角度
- **End Rotation** - 回転後の角度。赤い矢印で表示される。
- **Flipper Rotation Time** - 回転する時間。小さい値（0.04）が望ましい。
- **Flipper Side** - 右側のフリッパーか左側のフリッパーにアサインする。それぞれ右と左のSHIFTキーでプレイヤーが操作する。
- **Flipper Color** - フリッパーのプラスチック部分の色
- **Rubber Color** - フリッパーのゴム部分の色

BP_Slingshot



ボールを跳ね返す三角のバンパーです。だいたい台の下方のフリッパーの近くにあります。

- **Flip X** - 台の左側に配置するときに Flip X で反転できる
- **Power** - ボールを跳ね返す力

BP_PopBumper

ポップアクションでボールを跳ね返すバンパーです。

- **Flash Color** - 光るときの色
- **Test Lightup** - ライトアップされた状態をテストするためのスイッチ
- **Pop Speed Multiplier** - ボールを跳ね返すパーツが動くスピード
- **Bumper Cap Material** - バンパーのキャップに割り当てるマテリアル
- **Bumper Cap Texture** - pop_bumper_capというマテリアルと一緒に使用すると、このテクスチャーのアルファ チャンネルはステンシルのようなロゴなどを追加可能

古いプロパティ（エディターで表示されない）

- **Fake It** - 移動するパーツの実際の物理シミュレーションの代わりに、ボールに直接力を与えます。
- **Power** - ボールを跳ね返す力

BP_Post



ゴムのリングが付いている単純なポストです。台の特定エリアをブロックするために便利です。特に、ボールがよく引っかかって動かなくなるところがあったら、そこに入らないようにポストを使用してみてください。

- **Post Color** - プラスティックのベースの色
- **Light Color** - ポスト内にあるライトの色。現在、このライトはいつもオンになっている
- **Rubber Color** - ゴムのリングの色

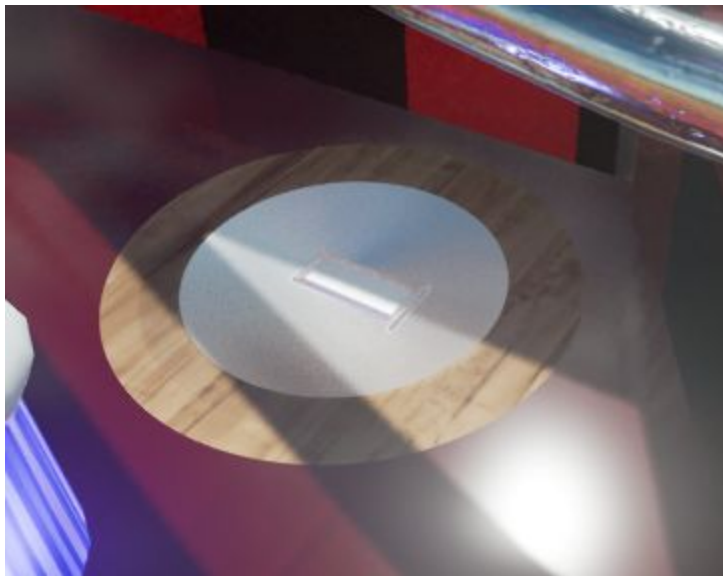
BP_PostRail



上記のポストと同じですが、2つのポストにゴムのバンドが付いています。長さの変更ができるので、ボールを跳ね返す壁として使用できます。

- **Post Color** - プラスティックのベースの色
- **Light Color** - ポスト内にあるライトの色。現在、このライトはいつもオンになっている
- **Rubber Color** - ゴムのバンドの色
- **End Post Location** - 2つ目のポストの位置。3Dのウィジットで移動することも可能

BP_KickerHole



ボールをトラップする穴です。特定の時間が経過すると、ボールが穴から飛ばされます。

- **Trap Time** - ボールを飛ばすまでの時間（秒）
- **Kick Direction** - ボールを飛ばす方向を設定。3Dのウィジットで移動することも可能

BP_TargetSet



ドロップ ターゲットのセットです。ボールがこのターゲットにぶつかると、ターゲットが倒され、プレイヤーのスコアにポイントが追加されます。注意 : *Target Set* は以下の *TargetController* で管理されます。

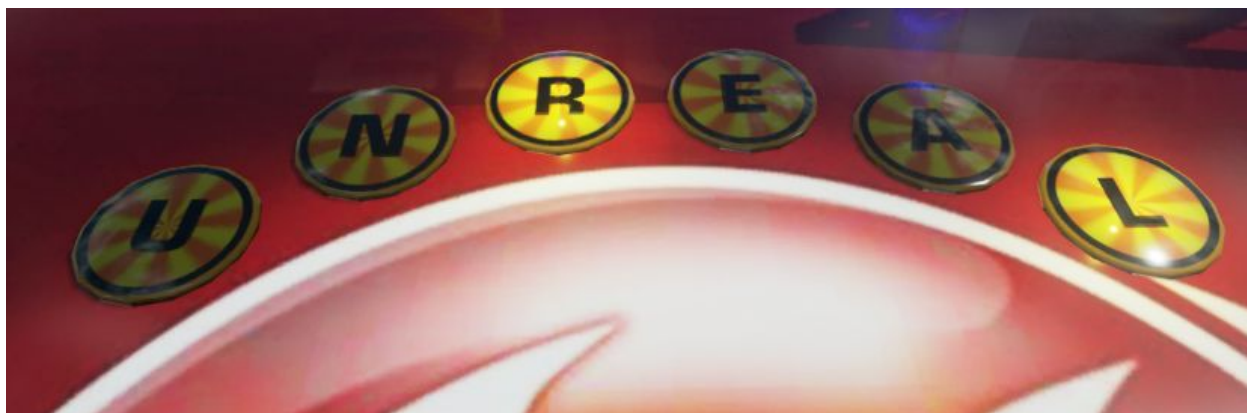
- **Number of Targets** - セットにあるターゲット数
- **Target Score Value** - ターゲットを倒すときのスコア
- **Unify Colors** - ターゲットの色パターンを一致させます（色パターンの説明は下記を参照）。ここにチェックを入れると、すべてのターゲットは、Target Data 0で指定した色パターンを使用します。
- **Target Data** - 各ターゲットのデータを持つ構造体（Struct）です。
 - **Target Style** - このリストで選択してアルファベットの文字をターゲットに表示させることができます。
 - **Drop Target Texture** - ターゲットで使用するテクスチャー。アルファ付きのテクスチャーであれば、後ろに *Base Color* が表示されます。アルファベットの文字を使用している場合に、文字はテクスチャーの上に表示されます。
 - **Base Color** - ターゲットの基本の色
 - **Target Color** - ターゲットアイコンの色（DropTargetTexture を使用しない場合のみ）
 - **Letter Color** - アルファベット文字の色
 - **Object To Activate** - ターゲットが倒されたときにこのアクターをアクティブにします（BP_IndicatorLight のライトをオンにするなど）

BP_TargetController

ターゲットセットを制御するために、ターゲットコントローラーを使います。全てのターゲットが倒れたらターゲットをリセットして、オプションで特別モードに入る処理はコントローラーで行われています。

- **Target Sets** - 管理する Target Set の配列。Target Set が台に存在する場合に、ドロップダウンにリストアップされます。

BP_IndicatorLight



台の表面に追加できる単純な丸いライトです。何らかの条件に基づいてオン／オフにすることが可能。倒したターゲット、スコアなどを表示するために便利です。

- **Light Color** - ライトの色
- **Indicator Material** - ライトのメッシュに使用されるマテリアル（デフォルトの M_TableLightBase を使用すると、以下の設定が使える）。
- **Icon Texture** - メッシュのテクスチャ
- **Activation Sound** - ライトがオンになるときに再生するサウンド
- **Test Lit** - ライトアップされた状態をテストするためのスイッチです（ゲーム開始時に、この設定が無視され、ライトがオフの状態から開始）。
- **Use Letter** - このリストでアルファベットの文字をライトに表示させることができます。ターゲットの組み合わせで、アルファベット文字付きのターゲットとライトのペアを簡単に作ることが可能。

BP_IndicatorRollover

BP_Indicator_lightに似ていますが、ボールがこの上に転がるとライトがオン／オフします。

- **Light Color** - ライトの色
- **Indicator Material** - ライトのメッシュに使用されるマテリアル（デフォルトの M_TableLightBase を使用すると、以下の設定を使用可能）
- **Icon Texture** - メッシュのテクスチャー
- **Activation Sound** - ライトがオンになるときに再生するサウンド
- **Test Lit** - ライトアップされた状態をテストするためのスイッチ（ゲーム開始時に、この設定が無視され、ライトがオフの状態から開始）
- **Use Letter** - このリストでアルファベットの文字をライトに表示することができます。ターゲットの組み合わせで、アルファベット文字付きのターゲットとライトのペアを簡単に作ることが可能。

BP_IndicatorSpecial



Tilt、マルチボールなどの特別モードの状態を表示するインディケーターです。テーブルに配置することで、PinballGameMode が自動的にそれを見つけて、特別モードがアクティベートされる時にオンに出来るように登録します。

- **Special Type** - （Tilt, マルチボールなどの）特別モードの種類のライトの設定
- **Light Color** - ライトの色
- **Indicator Material** - ライトのメッシュに使用されるマテリアル。（デフォルトの M_TableLightBase を使用すると、以下の設定を使用可能）
- **Icon Texture** - メッシュのテクスチャー

- **Activation Sound** - ライトがオンになるときに再生するサウンド
- **Test Lit** - ライトアップされた状態をテストするためのスイッチ（ゲーム開始時に、この設定が無視され、ライトがオフの状態から開始）
- **Use Letter** - このリストでアルファベットの文字をライトに表示することが可能。基本的に使用しない。

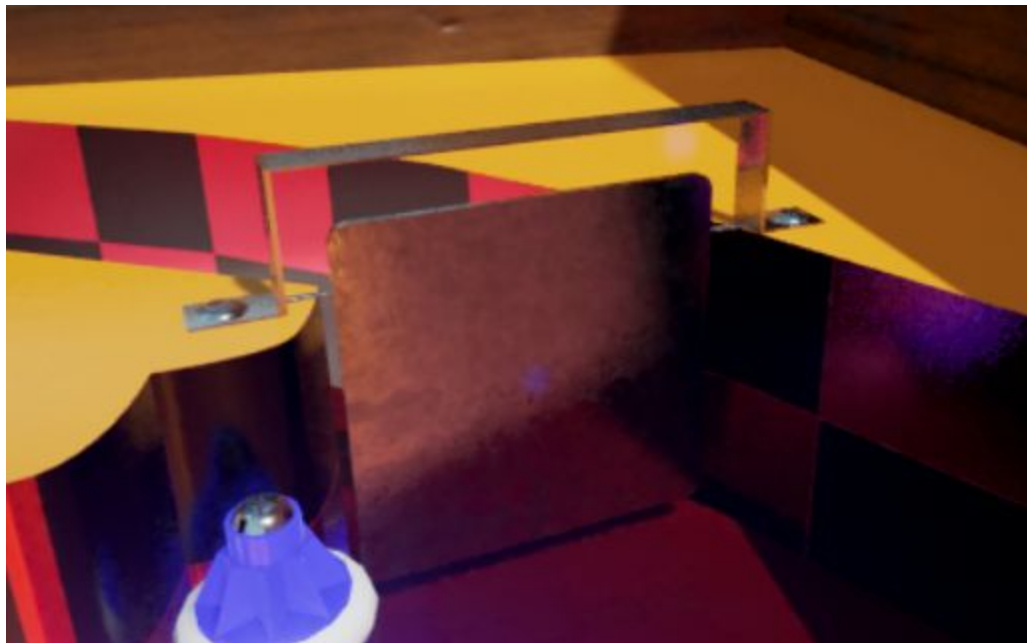
BP_BallChannel



通常、テーブルのプレーヤーから遠い所にある間隔をおいて配置されているバンパー。スコアに影響しない単なる壁です。

- **Count** - メッシュの数
- **Spacing** - メッシュの間の距離。ボールの直径は 100 ユニットですので、それより大きい数字を使ってください。

BP_Gate



一方通行のゲートです。赤い矢印はボールが通過できる方向を示します。現実のピンボール台で、これはシューターの出口のところによくあります。ボールをプレイエリアにシュートするときにこのゲートを通りますが、またシューターのところに戻らないようにボールをブロックします。

- **Bracket End Point** - 3D ウィジットでブラケットの大きさを調整
- **Bracket Height** - 床からの高さ（ネジが付いている部分）
- **Tile Width Multiplier** - ゲート部分の幅を調整する乗数。デフォルトのサイズは 100 ユニット
- **Gate Clearance Height** - 床とゲート下部分の間の距離。ゲートがよく動くように、少なくとも 10 ユニットぐらいに設定すると良い

BP_Spinner



単純なスピナーです。ボールが通るとクルクル回転します！

- **Bracket End Point** - 3D ウィジットでブラケットの大きさを調整
- **Bracket Height** - 床からの高さ（ネジが付いている部分）
- **Tile Width Multiplier** - スピナー部分の幅を調整する乗数。デフォルトのサイズは 100 ユニット
- **Spinner Texture** - スピナーのテクスチャー
- **Spinner Clearance Height** - 床とスピナー下部分の間の距離。ゲートがよく動くように、少なくとも 10 ユニットぐらいに設定すると良い。

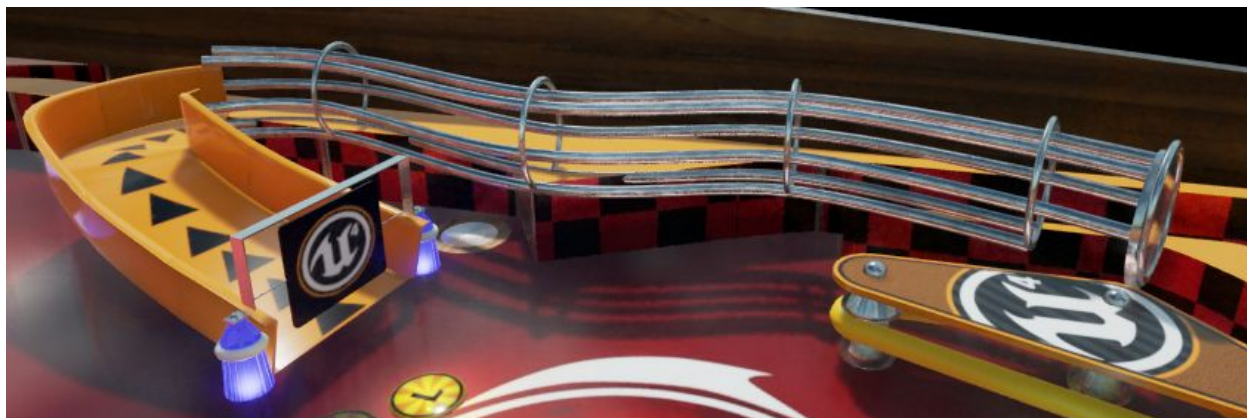
BP_WallActor



このブループリントで、ピンボール台にスプラインからできた壁を作れます。このプロジェクトでエディターのカスタマイズをしたので、詳細パネルでスプラインを 2D で編集することができます（下記のスプライン Actor の詳細パネルのセクションをご覧ください）。

- **Height** - 壁の高さ
- **Spline Wall Mesh** - 壁のメッシュ（基本的にデフォルトの分解された表面を変更する必要はない）
- **Cap Material** - 壁のトップ面で使用するテクスチャー
- **Use Zero Material For All Walls** - これにチェックを入れると、壁のすべてのセグメントは、スロット 0 のマテリアルを使用（下記を参照）
- **Side Materials** - 壁の各セグメントに使用するマテリアルの配列。壁を作るときに、番号付けされたマテリアルが自動的に適用されます。その数字はこの配列のインデックスを示します。

BP_RampActor



ランプのエディターでランプとレールをスプラインで簡単に実現出来ます（下記のスプライン Actor の詳細パネルのセクションを参照）。

ランプのプロパティ:

- **Start Height** - ランプの入り口の高さ。通常、0に設定。
- **End Height** - ランプ坂の高さ。Start から end までの距離（坂の長さ）は *Slope Section Count* のパラメータで設定。
- **Start Width** - ランプの入り口の幅。入り口が広いほど、ボールが入りやすくなります。
- **End Width** - ランプ終端の幅。レールの部分を除いて、ランプの幅はスプラインの長さに沿って線形補間されます。
- **DefaultLength** - ランプのデフォルトの長さ。
- **Wall height** - ランプの壁の高さ。
- **Slope Section Count** - 坂の *Start Height* から *End Height* まで補間するスプラインセクションの数。この値が小さいほど、急な坂になります。大きい数字だとたくさんのセクションで補間するため、緩やかな坂になります。
- **Cap End** - このプロパティにチェックを入れると、ランプ終端に合わせて穴がある末端部が追加されます。
- **Ramp Mesh** - ランプのスタティックメッシュ。
- **Ramp Material** - ランプのマテリアル。

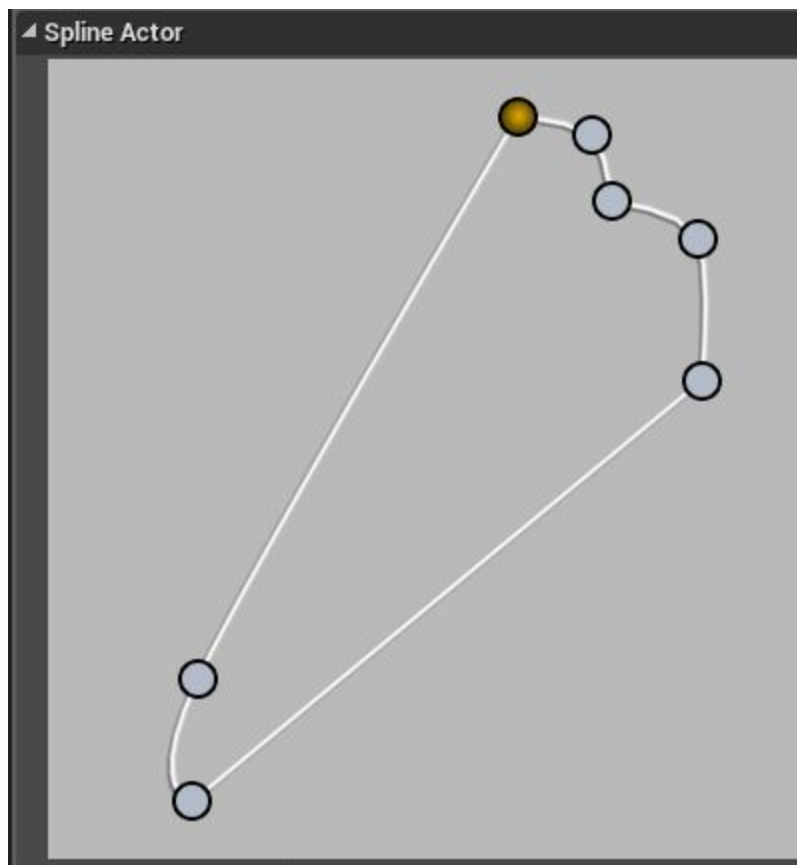
レールのプロパティ:

- **Rail Mesh** - レールの部分で使うメッシュ
- **Rail Section Count** - レールに変換するスプラインセクションの数。スプラインの後ろから数えています。最初に幾つかのランプセクションを用意してからレールにトランジションするとうまく機能します。

- Rail Material - レールに適応するマテリアル

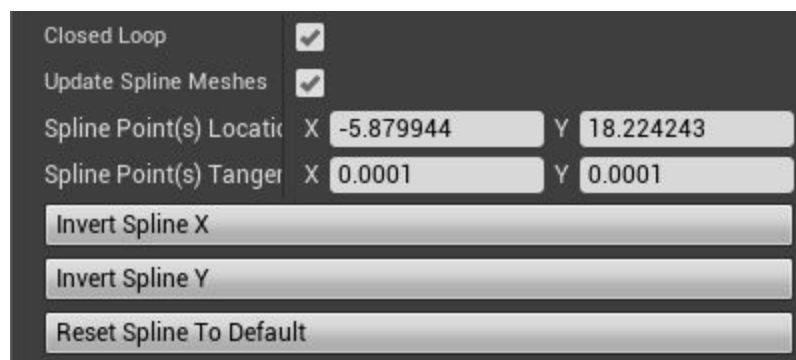
スプラインActorの詳細パネル (BP_WallActor & BP_RampActor)

壁やランプの形を定義するスプラインの編集プロセスをより早くスムーズに行えるように、そしてより精度が高い制御や新しい機能を提供する為に、スプライン Actor の詳細パネルのカスタマイズを追加しました。



まずは、スプラインの 2D のオーバーヘッドビューがあります。ここでスプラインポイントを 2D 面でドラッグしたり、ポイントを右クリックしてカーブタイプを変更またはポイントを削除することが出来ます。複数選択と取り消し／やり直しにも対応しています。ポイント間のスプラインセクションを右クリックすることで、新しいスプラインポイントを追加出来ます。

その下に幾つか追加したチェックボックスやボタンがあります：



- **Closed Loop** - スプラインが線であるかループであるか
- **Update Spline Meshes** - かなり複雑なスプラインがある場合、このチェックボックスのチェックを取ることで、コンストラクションスクリプトを毎フレーム実行せずにスプラインを編集することが出来ます。編集が終わったら、再度チェックを入れると、メッシュの変更が適用されます。
- **Spline Point(s) Location** - 現時点で 2D ビューで選択しているスプラインポイントの位置を手動で編集出来ます。
- **Spline Point(s) Tangent** - 現時点で 2D ビューに選択しているスプラインポイントのタンジェント（接線）を手動で編集出来ます。
- **Invert Spline X** - スプラインを X 方向でインバートする
- **Invert Spline Y** - スプラインを Y 方向でインバートする
- **Reset Spline to Default** - スプラインをデフォルト状態に戻す

ピンボールの特別コンポーネント

ピンボールの複数のパーツで必要になっていた機能を持つブループリントコンポーネントを2つ制作しました。スコアコンポーネントは、プレイヤーのスコアにポイントを加える処理をします。特別モードコンポーネントは、ある条件を達成すると特別モードに入る処理をします。

オブジェクトをレベルに配置してから、こうしたコンポーネントを詳細パネルで選択して、設定を変えることが出来ます。

スコアコンポーネント

このコンポーネントは単にプレイヤーのスコアにポイントを加えます。例えば、バンパーに衝突するかターゲットを倒す時に、そのブループリントがこのスコアコンポーネントにある関数を呼ぶ事でポイントを加えます。スコアの乗数は GameMode というクラスで管理されます。

- **Point Value** - スコアに加えるポイントの数

特別モードコンポーネント

ゲームプレイの条件に応じて特別モードに入ります。例えばターゲットセットの中にある全てのターゲットを倒せば、特別モードコンポーネントがマルチボールモードに入ったり、エクストラボールを与えることなどが出来ます。

- **Special Mode** - どの特別モードに入るかという設定。以下の特別モードを含みますが、E_SpecialModeというエニューメレーターに項目を追加することで更に追加出来ます。以下の特別モードの多くのモードに対応するライトがあって、そのライトをBP_Indicator_Specialというクラスでテーブルに追加することが出来ます。対応するライトがテーブルに存在すると、特別モードに入る時に GameMode のクラスが自動的にそのライトをオンにします。
 - **None** - 何もしない
 - **Trigger Actor** - 好きなアニメーションやエフェクトなどの任意の物をトリガーする汎用のモード。ブループリントを作成して PinballInterface というインターフェースを実装して、EventActivate というイベントでトリガーしたいアクションをトリガー出来ます。そのブループリントを特別モードの ActorToTrigger というプロパティにアサインすることで特別モードに入るとトリガーされます。
 - **Tilt** - プレーヤーがテーブルを揺らしすぎた場合にフリッパーがロックされるモード。デフォルトのプレーヤーコントローラーでは、3回テーブルを揺らすとティルトモードに入ります。
 - **Ball Save** - このモードが有効になっている時に、アウトホールに落ちるボールがセーブされて、自動的にテーブルに再度打ち出されます。デフォルトのゲームモードでは、プレイ開始後、30秒が経過するまでボールセーブモードが有効になります。
 - **Multiball** - マルチボールモードをトリガーします。現時点のコンフィグレーションでは追加で3つのボールをテーブルに打ち出して、30秒が経過するまでボールセーブモードを有効にします。
 - **Extra Ball** - プレーヤーにエクストラボールを与えます
 - **Score 2x** - スコアの乗数を2倍にします*
 - **Score 3x** - スコアの乗数を3倍にします
 - **Score 4x** - スコアの乗数を4倍にします*
 - **Score 6x** - スコアの乗数を6倍にします
 - **Progressive Multiplier** - このモードは毎回スコアの乗数を次の段階に上げます
- **Retriggerable** - このモードを 1 ゲームに複数回をアクティベートしたい場合にチェックを入れてください。チェックを外すと、1 ゲームに 1 回のみアクティベートされます。
- **ActorToTrigger** - TriggerActorという特別モードでトリガーしたいActorを設定出来ます。



*現在の設定で、スコアの乗数の3倍と6倍の場合に、ボールからパーティクルのトレールが出るようになります。

テーブルメニューに自分のテーブルを追加

テーブルのリスト

個別のテーブルについての情報は DataTables>TableList に保存されています。このリストに新しいテーブルに関する情報を追加することでテーブルのロードのメニューに表示させることができます。

- **TableAssetName** - ピンボールのテーブルのマップ (.umap)ファイル名
- **TableDisplayName** -ピンボールテーブルの名前（ディスプレイのみ）
- **TableDescription** -テーブルを説明するテキスト
- **TableThumbnail** - メニューに表示されるテーブルのサムネイル画像

既存の問題点

ライティングのビルド

時々、あるスプラインメッシュとシャドウをキャストするスタティックもしくはステーションナリーライトの組み合わせで、「Maps need lighting rebuilt」というエラーメッセージが表示されてライティングのビルドが失敗することがあります。

この問題を確実に再現する方法がまだ見つかっていないため、現時点では解決方法を調べることは出来ません。

この問題が発生したら、問題になっているライトのシャドウをキャストしないように設定するか、ライトを移動してみてください。

Play in Standaloneで発生するスプラインメッシュのコリジョンの問題

Play in Standalone game では、スプラインメッシュのコリジョンが適切に機能していません。この問題は 4.9 と 4.10 で発生しますが、将来の 4.11 のリリースで修正されます。

パッケージ化ゲームでは現在のバージョンでもスプラインメッシュのコリジョンが正常に機能しています。