Pinball Construction Kit

November 2015

Joe Conley & Rob Gray
Epic Games Japan

Overview

The Pinball Construction Kit is a sample project created for Unreal Fest 2015 in Japan. The goal was to not only make a pinball game in UE4, but to create a framework for making multiple pinball tables through the use of blueprint and editor customization.

We hope this is a valuable learning resource and also look forward to seeing what the community can do with it. If you design your own pinball table, please post it on the forums for others to enjoy.

Getting Started

1. The StarterTable map file can be used to start a new table
2. All Tables must include a BP_Plunger to fire balls and a BP_Drain to register lost balls.
3. Then just drop in targets, bumpers, lights, ramps, etc. and have fun!

Creating Your Own Pinball Table

We've designed this project to make it easy to drop in pinball machine features to quickly build a pinball table of your own design. The project includes many pre-made pinball machine parts that can be placed in your table and customized, allowing quick creation of custom tables that can be played without any coding or other work.

But for those of you who want to extend the Blueprints and add your own features, here's a description of how the main game part of the project is set up:

MyPinballGameMode

The GameMode contains most of the basic rules and functionality of the pinball game. We've created a basic framework here, but feel free to extend from this class or create your own to handle special rules of your own table. The default MyPinballgameMode does the following things:

- When the game starts, it finds and assigns a BP_Plunger to control shooting balls.
- It also looks for any BP_Indicator_Special objects and saves them to be activated when the table goes into Special Modes. These are special indicator lights that designate things like Tilt state, Multiball, Score Multipliers, etc.
- It adds the UMG scoreboard HUD to the game
- By default, it plays a startup sound (music) and does some fancy animation through all the lights on the table. After this, Ball 1 is loaded for play.
- By default, it starts the game with a 30-second Ball Save mode. While this mode is active. any ball that falls into the gutter will be saved and automatically fired back onto the table.

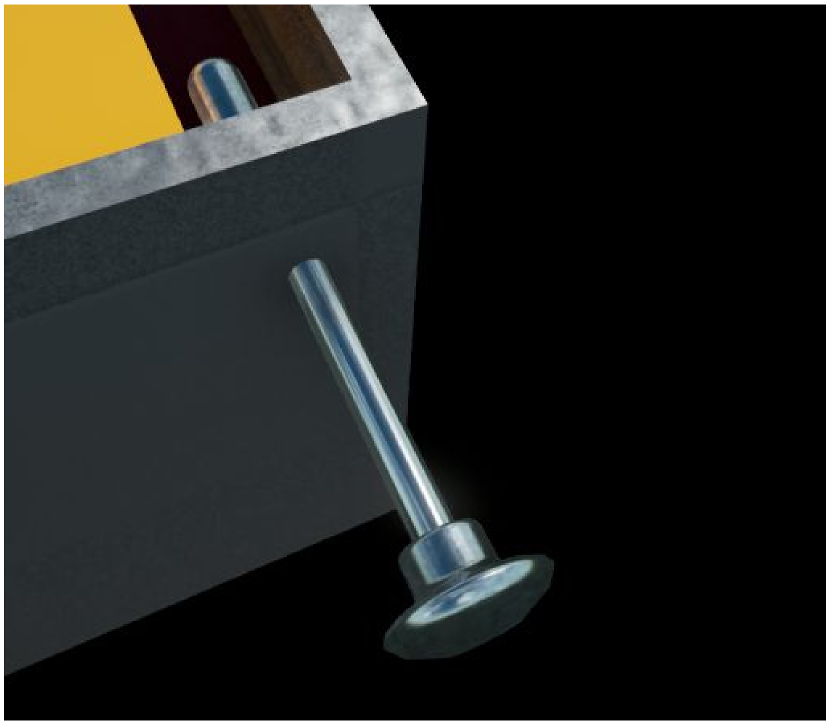
PinballPlayerController

The default Player Controller will find any BP_Flipper objects placed on the table and assign them to be activated with the Right and Left Shift keys. It also handles using Left and Right Ctrl keys to nudge the table. When the table is nudged too many times, it will go into Tilt mode and deactivate usage of the filppers.

Pinball Table Parts

The Pinball Construction Kit includes many ready-to-use and customizable pinball machine features that were created using Blueprint. Drag and drop these into your own table, adjust a few settings and quickly make a table of your own!

BP_Plunger



This is the main plunger that fires balls and must be added to any table to make it work.

BP_AutoPlunger

These are plungers typically found in the left and right gutter channels of the table. When the ball is lost in the gutters, these will automatically fire to save the ball and put it back in play.

- **NumberOfSaves -** The number of times the Plunger will activate to save the ball

BP_Drain

This is required on every table and should be placed behind the flippers to register lost balls.

BP_Flipper

MULTI BALL
TILT
BALL SAVE

Pinball machine flippers. When the game starts, these are automatically registered to the PlayerController and can be controlled with Left and Right Shift keys.

- **Start Rotation -** Angle of flipper in rest position
- **End Rotation -** Angle of flipper in hit position, represented by the red arrow
- **Flipper Rotation Time -** Time it takes flipper to rotate. Small values (0.04) should work well
- **Flipper Side -** Assigns the flipper as a right flipper or left flipper, to be controlled with the right and left shift keys, respectively
- **Flipper Color -** Sets the color of the plastic part of the flipper
- **Rubber Color -** Sets the color of the rubber part of the flipper

BP_Slingshot

These are the powered, triangular bumpers that are typically at the close-end of the table, near the flippers.

- **Flip X -** Use Flip X when placing a slingshot on the left side
- **Power -** The strength at which the slingshot repels the ball

BP_PopBumper

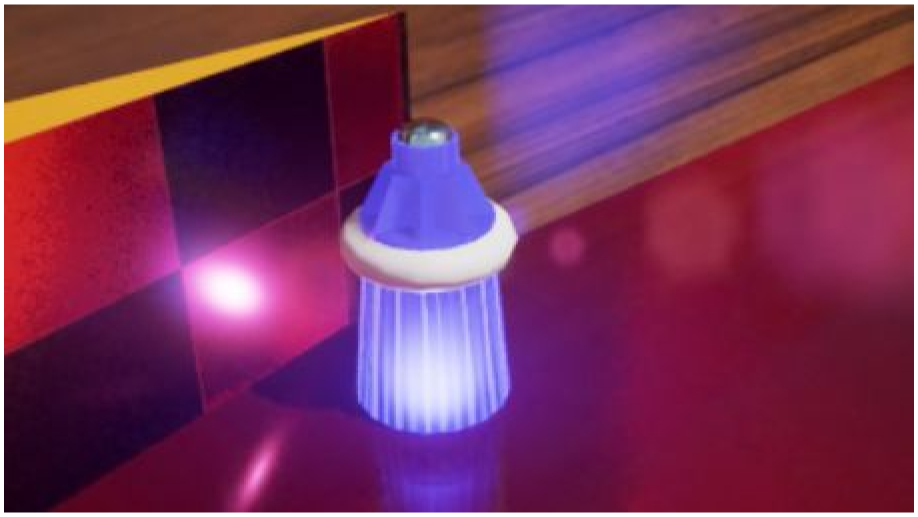
Classic, pinball pop bumpers.

- **Flash Color -** Sets the color of the bumper when it lights up
- **Test Lightup -** Editor toggle to test the bumper's lit state
- **Pop Speed Multiplier -** Speed at which the bumper plate moves to collide and repel the ball
- **Bumper Cap Material -** Material for the bumper cap design
- **Bumper Cap Texture -** Used with pop_bumper_cap material, the alpha of this texture can be used to add a stenciled logo, etc.

Deprecated properties:

- **Fake It -** Whether to use an impulse in the ball direction rather than actual physical collision to repel the ball
  Power - The strength of the impulse at which the bumper repels the ball

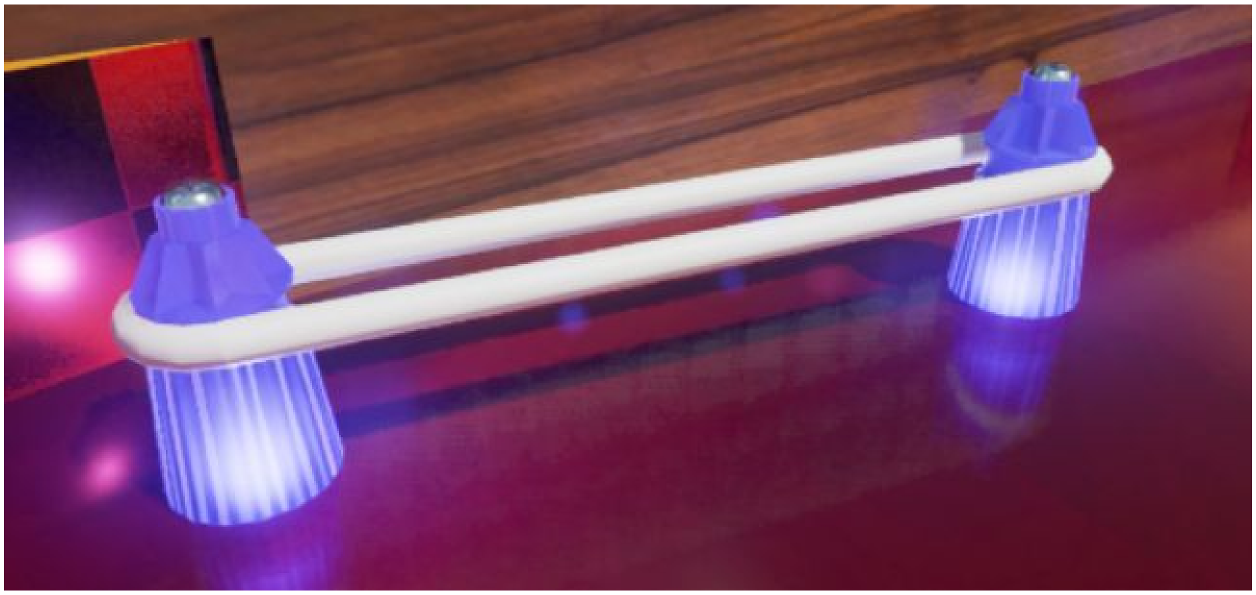
BP_Post



A simple pinball post with a rubber ring that repels the ball. Useful for blocking areas of the table, especially where you don't want a ball to get stuck.

- **Post Color -** Sets the color of the plastic post base
- **Light Color -** Sets the color of the light inside the post. Currently these lights are always on.
- **Rubber Color -** Sets the color of the post's rubber ring.

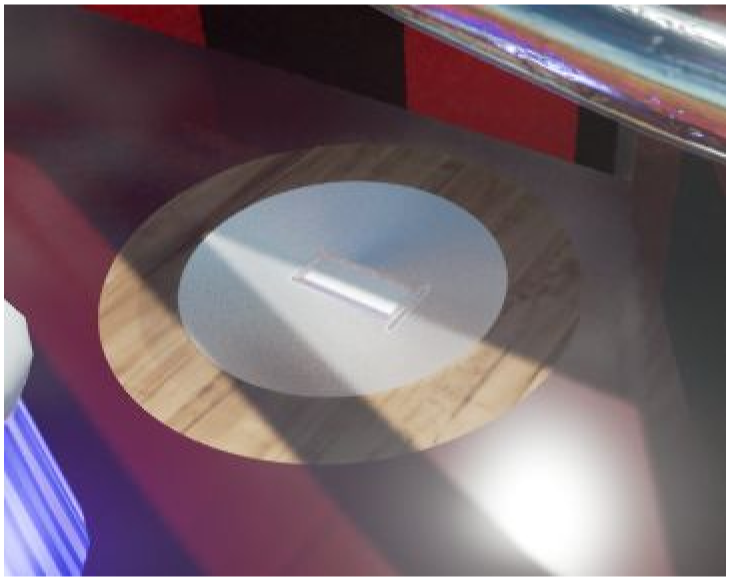
BP_PostRail



A set of 2 posts, which a rubber band wrapped around them. Can be adjusted for size and used as a bouncy wall.

- **Post Color -** Sets the color of the plastic post bases
- **Light Color -** Sets the color of the light inside the posts. Currently these lights are always on.
- **Rubber Color -** Sets the color of the posts' rubber band
- **End Post Location -** The location of the 2nd post in relation to the 1st. This can also be set with the 3D widget

BP_KickerHole



A hole that traps the ball and then spits it out.

- **Trap Time -** How many seconds the ball stays in the hole before being kicked out
- **Kick Direction -** Sets the direction in which to kick the ball out of the hole. Can also be set with the 3D widget

BP_TargetSet

A set of drop targets. When the ball hits these, they are knocked down and score is added. *Note: Target Sets are controlled with a TargetController (see below).*

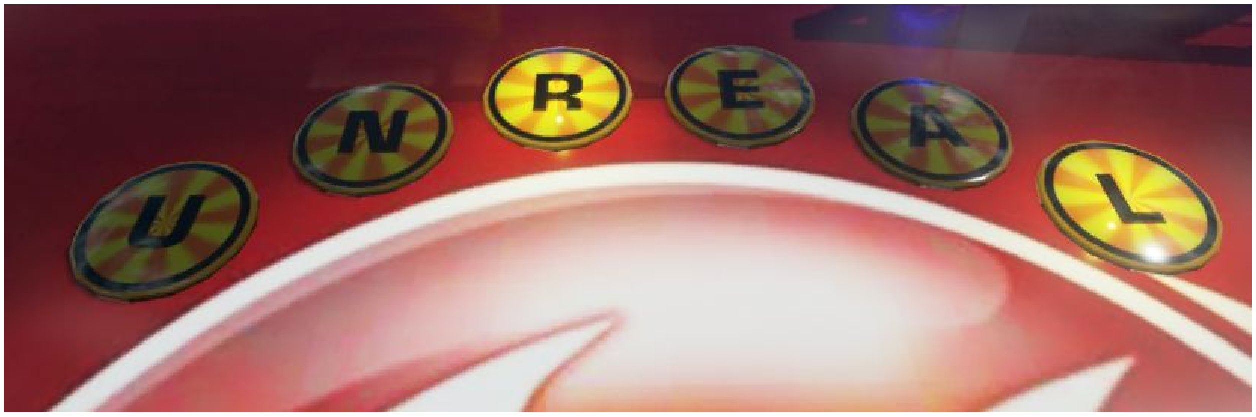
- **Number of Targets -** The number of targets in the set
- **Target Score Value -** Sets the point value of the targets
- **Unify Colors -** Use the same color scheme for all targets (see below). This will cause all targets to use the color scheme defined in Target Data 0
- **Target Data -** A struct that holds data for each target
  - **Target Style -** A letter of the alphabet can be easily assigned to the target by choosing from this dropdown
  - **Drop Target Texture -** Base texture for the target. If using a texture with alpha, the *Base Color* will show behind it. Letters can be shown over the base texture
  - **Base Color -** The base color of the plastic target
  - **Target Color -** The color of the bullseye target icon (Only when not using a DropTargetTexture)
  - **Letter Color -** The color of the alphabet letter on the target
  - **Object To Activate -** Object to activate when the target is downed (typically used to turn on an *indicator light*, etc.)

BP_TargetController

A Target Controller is used to control one or more Target Sets. The controller handles resetting the targets when all are downed, and optionally, starting a *Special Mode.*

- **Target Sets -** The Target Sets controlled by the controller. If any Target Sets are on the table, they can be selected from the dropdown.

BP_IndicatorLight



A simple round light on the table that can be turned on/off by other parts. Typically used to show scoring, downed targets, etc.

- **Light Color -** Sets the color of light
- **Indicator Material -** The material used for the light. (M_TableLightBase works with the other settings below)
- **Icon Texture -** The texture to use for the light
- **Activation Sound -** Sound to play when the light is turned on
- **Test Lit -** Editor-only toggle to test the lit state of the light. (When the game starts, the light ignored this setting and starts in the off state)
- **Use Letter -** Add a stenciled letter texture to the light. This can be used in conjunction with letters on targets to make matched pairs or target and light.

BP_IndicatorRollover

Similar to the Indicator light, but turns on/off when the ball rolls over it.

- **Light Color -** Sets the color of light
- **Indicator Material -** The material used for the light. (M_TableLightBase works with the other settings below)
- **Icon Texture -** The texture to use for the light
- **Activation Sound -** Sound to play when the light is turned on
- **Test Lit -** Editor-only toggle to test the lit state of the light. (When the game starts, the light ignored this setting and starts in the off state)
- **Use Letter -** Add a stenciled letter texture to the light. This can be used in conjunction with letters on targets to make matched pairs or target and light.

BP_IndicatorSpecial

MULTI BALL
TILT
BALL SAVE

A special class of Indicator lights used to designate *Special Modes,* such as *Tilt, Multiball,* etc. When placed on the table, the PinballGameMode will automatically find a register these so that they can be turned on when a *Special Mode* is activated.

- **Special Type -** Sets the type of special light to use. (Tilt, Multiball, etc.)
- **Light Color -** Sets the color of light
- **Indicator Material -** The material used for the light. (M_TableLightBase works with the other settings below)
- **Icon Texture -** The texture to use for the light
- **Activation Sound -** Sound to play when the light is turned on
- **Test Lit -** Editor-only toggle to test the lit state of the light. (When the game starts, the light ignored this setting and starts in the off state)
- **Use Letter -** Add a stenciled letter texture to the light. This can be used in conjunction with letters on targets to make matched pairs or target and light.

BP_BallChannel

This is just a set of spaced bumpers typically found near the far end of the table. They are simply walls with no scoring value.

- **Count -** The number of blockers to add
- **Spacing -** The distance between blockers. The default diameter of the ball is 100 units, so the spacing should be larger than that to allow the ball to pass through

BP_Gate

This is a one-way gate that can be placed on the table. The arrow shows which direction the ball may pass through. On a real pinball table, these are often found at the end of the plunger channel. When the ball is shot with the plunger, the gate allows the ball to go through into the playfield, but prevents it from going back into the plunger channel.

- **Bracket End Point -** Use the 3D widget to place the opposite side of the gate bracket
- **Bracket Height -** Used to adjust the height of the gate
- **Tile Width Multiplier -** A multiplier to change the width of the gate tile. The default width is 100 units.
- **Gate Clearance Height -** Height of the gate from the ground. Make sure this is at least 10 units to get good swing

BP_Spinner

A basic spinner. The ball goes through. It spins!

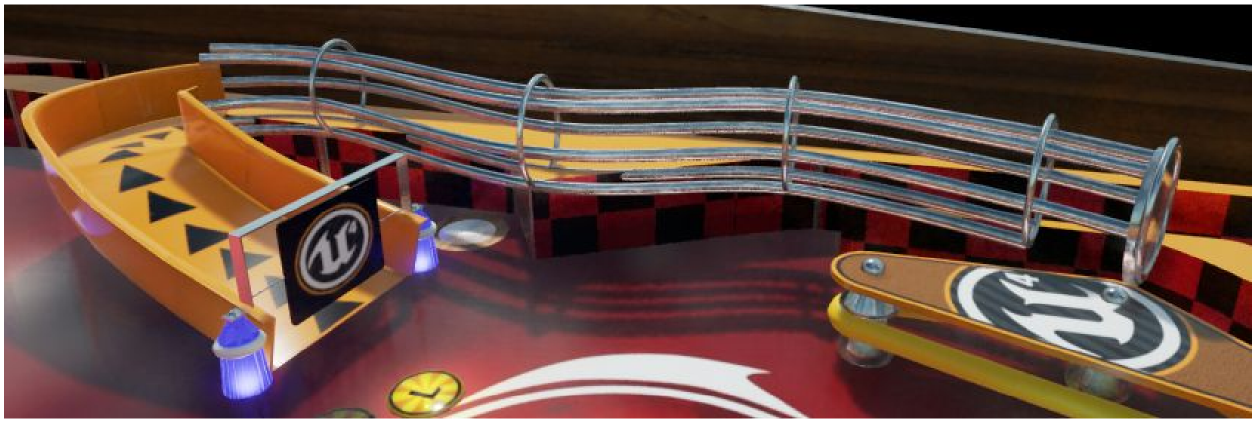
- **Bracket End Point -** Use the 3D widget to place the opposite side of the spinner bracket
- **Bracket Height -** Used to adjust the height of the bracket
- **Tile Width Multiplier -** A multiplier to change the width of the spinner tile. The default width is 100 units.
- **Spinner Texture -** Texture to set on the spinner
- **Spinner Clearance Height -** Height of the spinner tile from the ground surface. Make sure this is at least 10 units to get a good spin

BP_WallActor

This Blueprint is for making spline-based walls on your table. Through editor customization, the Blueprint features a 2D spline editing window.

- **Height -** The height of the wall
- **Spline Wall Mesh -** The mesh to use for the walls
- **Cap Material -** The material to use for the top of the walls
- **Use Zero Material For All Walls -** When this is checked, all wall segments will use the material in slot 0 of Side Materials (see below)
- **Side Materials -** The materials used for each segment of the wall. By default, the walls are assigned a numbered material that corresponds to each index in this array

BP_RampActor



The ramp editor makes it easy to create ramps and rails with splines.

Ramp Properties:

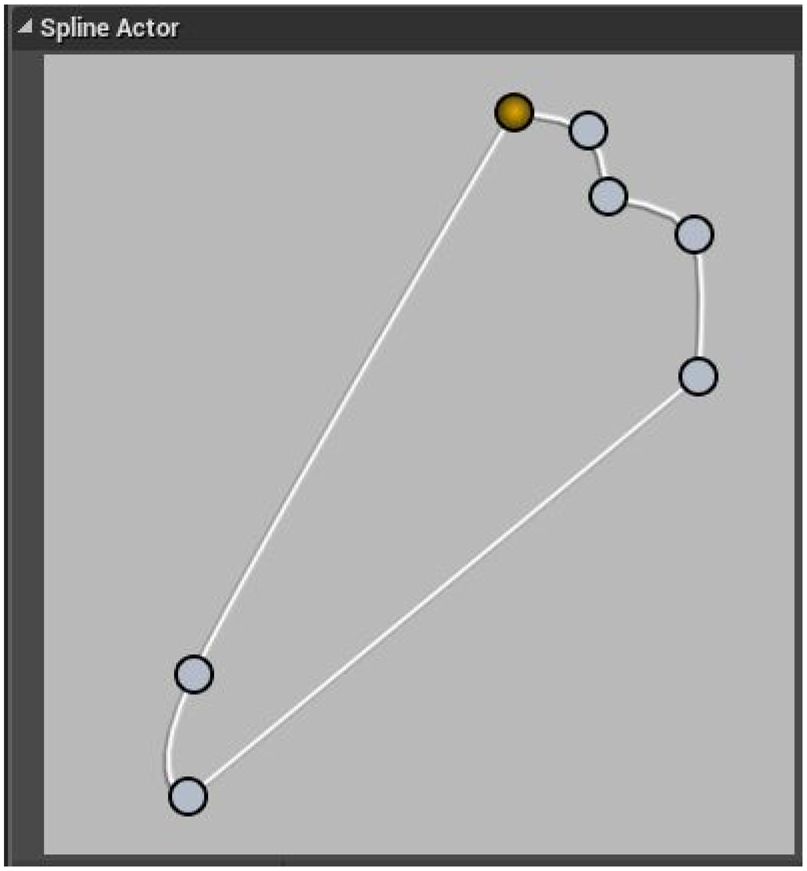
- **Start Height -** The start height of the ramp. Typically 0
- **End Height -** The end height of the ramp. The length between start and end heights is determined by *Slope Section Count*
- **Start Width -** The starting width of the ramp. A wider start width makes it easier for a ball to enter the ramp
- **End Width -** The end width of the ramp. The width is linearly interpolated along the length of the spline, except for rail segments
- **DefaultLength -** The length of the default ramp
- **Wall height -** The height of the ramp's walls
- **Slope Section Count -** The number of ramp sections to use to interpolate from *Start Height* to *End Height.* A smaller number will use less sections and create a steeper slope. A larger number creates a more gradual slope
- **Cap End -** When this is checked, an end piece with a hole in the bottom is added at the end of the ramp
- **Ramp Mesh -** The Static Mesh to use for the ramp
- **Ramp Material -** The material to use for the ramp

Rail Properties:

- **Rail Mesh -** The mesh to use for the rails
- **Rail Section Count -** The number of spline sections to turn into rails. Having a few sections of ramp at the start and then transitioning to rails works best. This is counted from the end of the ramp.
- **Rail Material -** The material to apply to the rails

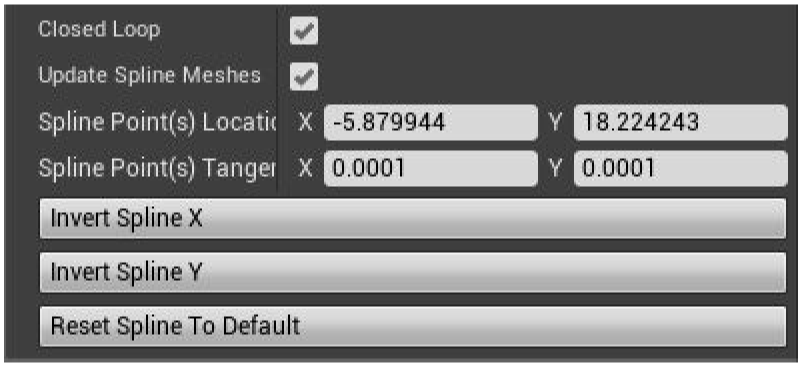
Details Panel for Spline Actors (MyWallActor & MyRampActor)

To make the editing process for the splines that define the shape of the wall and ramp actors quicker and easier, and to provide a little more precision and new features, we added a details customization for Spline Actors.



First, there is a 2D overhead representation of the spline. Here you can drag points around in 2D, or right click on them to change their curve type, or remove them. It also supports multiple selection of points, and undo/redo. You can also right click on sections of the spline to add new spline points.

Below that, there are a few additional checkboxes, buttons, etc.


Closed Loop ✓
Update Spline Meshes ✓
Spline Point(s) Locatic X -5.879944 Y 18.224243
Spline Point(s) Tanger X 0.0001 Y 0.0001

- **Closed Loop -** Whether the spline is a line, or a closed loop
- **Update Spline Meshes -** If you have a particularly complicated spline, you can uncheck the "Update Spline Meshes" button to allow the spline to be modified without the construction script running every frame, and then re-enable to apply the changes to the meshes.
- **Spline Point(s) Location -** Manually set values for spline point location for the spline points currently selected in the 2D view
- **Spline Point(s) Tangent -** Manually set values for spline point tangent for the spline points currently selected in the 2D view
- **Invert Spline X -** Invert spline in the X direction
- **Invert Spline Y -** Invert spline in the Y direction
- **Reset Spline to Default -** Reset the spline to the default

Special Pinball Components

Two Blueprint components were created to handle functionality that is needed for many of the individual pinball parts. The Score Component handles adding points to the player's score. The Special Mode Component handles triggering special modes of the game when conditions are met.

After placing an object, you can select these components in the Details panel and change their settings.

Score Component

This component simply adds points to the player's score. For example, when a bumper is hit or a target is downed, those blueprints call a function in this component to add points. Score multipliers are handled in the GameMode class.

- **Point Value -** The number of points to add to the score

Special Mode Component

This component handles the triggering of special modes based on some condition in gameplay. For example, when a full set of targets is downed, the Special Mode Component can start Multiball Mode or award an Extra Ball.

- **Special Mode -** Sets which Special Mode to start. The following modes are included in the dropdown, but more can be added by adding entries to the E_SpecialMode enumerator. Many of these Special Modes have corresponding lights that can be added to the table with the BP_Indicator_Special class. If a corresponding light exists, the GameMode will automatically turn it on when the mode is activated.
    - **None -** Do nothing special
    - **Trigger Actor -** An all-purpose mode to trigger animations, effects, and basically anything you want. Simply create a Blueprint and implement the PinballInterface. On EventActivate, trigger any actions that you wish. This Blueprint should be assigned to the *ActorToTrigger* property.
    - **Tilt -** A mode that locks the flippers when the player has nudged the table too much. The default PlayerController is set up to start Tilt mode after 3 nudges.
    - **Ball Save -** When this mode is active, any ball that drops into the outhole is saved and automatically fired back onto the table. The default GameMode starts the game with 30 seconds of Ball Save Mode.
    - **Multiball -** This triggers Multiball mode which is currently configured to shoot 3 extra balls onto the table. By default, it also turns on Ball Save for a short period of time.
    - **Extra Ball -** This awards an extra ball to the player.
    - **Score 2x -** Increases the score multiplier by 2
    - **Score 3x -** Increases the score multiplier by 3*
    - **Score 4x -** Increases the score multiplier by 4
    - **Score 6x -** Increases the score multiplier by 6*
    - **Progressive Multiplier -** This mode will cycle through the score modes, progressively increasing the multiplier
- **Retriggerable -** Check this if you want the special mode to be triggered multiple times. Otherwise, it will be triggered only once per game.
- **ActorToTrigger -** Assign an actor to be used with *Trigger Actor* special mode

**Currently, 3x and 6x multiplier will cause a particle trail to be emitted from all balls on the table*

Adding your table to the table menu

Table Lists

Individual table data is stored in DataTables>Table List. Adding information to this list will allow the table to be displayed on the table load menu.

- **TableAssetName -** This is the name of the pinball table map
- **TableDisplayName -** The name of your pinball table
- **TableDescription -** Text to describe your table
- **TableThumbnail -** A thumbnail image of your table to display on the menu

Known Issues

Building Lighting

Sometimes certain combinations of Spline Meshes and Static or Stationary Lights that cast shadows can cause the lighting build to fail with the messages "Maps need lighting rebuilt".
We haven't been able to reliably reproduce the issue, so we're unable to look into fixing it at this time.

If you run into this problem, disable shadows on the lights or try repositioning the lights.

Play in Standalone Collision Issue

Spline Mesh collision is not working properly in Play in Standalone game mode.
This issue occurs in 4.9 & 4.10, but will be fixed in 4.11
Collision works properly in a packaged game in current versions.