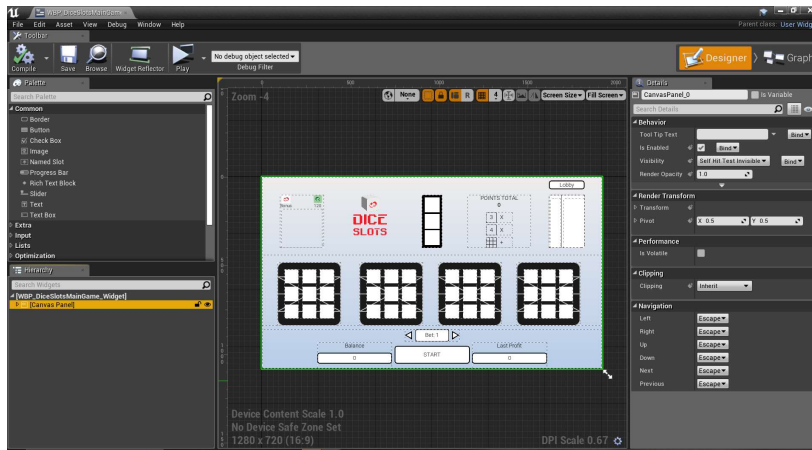


Dice Slots Blueprint Documentation

This documentation will cover the steps you would have to take in order to setup your own Dice Slots design using the Dice Slots Blueprint project.

Designing a new Dice Slots version

Assuming you have the assets you want to use for your dice slots game, such as button designs, dice, etc. first step you will have to do inside unreal is creating a new blueprint inheriting from `WBP_DiceSlotsMainGame_Widget`.



Next step is opening up the newly created class and also opening the `WBP_DiceSlotsMainGame_Widget` class itself. Select and copy the parent widget from `WBP_DiceSlotsMainGame_Widget`.

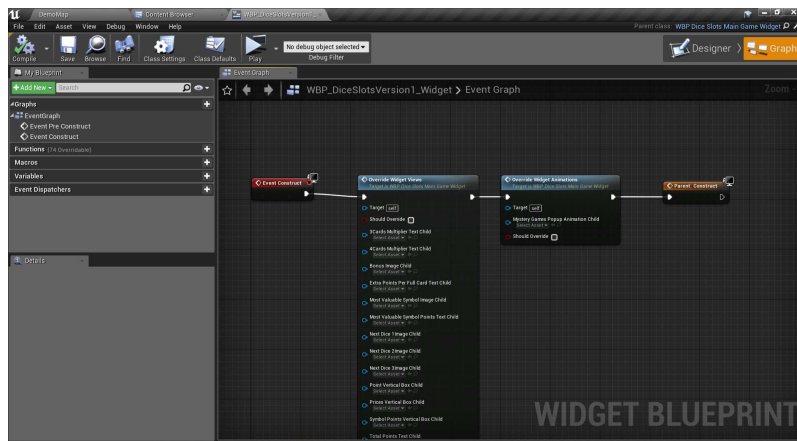
Next step is pasting the `MainCanvas` to your newly created widget class. At this point you don't need the `WBP_DiceSlotsMainGame_Widget` class so you can close that. Now if you try to compile the newly created child widget which you copied into, you'll get a lot of compile errors. That's because some of the widgets inside the widget are variables, and their names have already been occupied by the parent class which you copied from. In order to get past the compile errors, all you gotta do is just going through all elements in the widget's hierarchy tree, and rename those which are marked as variables. Non-variable widgets don't need to be renamed. I suggest just adding an underscore at the end of their names so you would avoid later confusions.

Later on, you can customize the widgets as you desire to get the looks you want. Now inside the child widget's blueprint you need to call a function called "Override Widget Views", one input variable of the function is called "Should Override" which you need to set to true, and there are a bunch of widget type input variables which you should connect to the relevant widget elements you copied and renamed in the child widget blueprint.

Also there's an animation in the parent widget. In order to have that, create an animation in the child widget, and as for the tracks, you can copy the tracks from parent blueprint and paste into child blueprint.

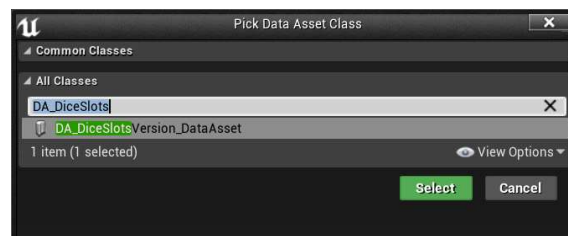
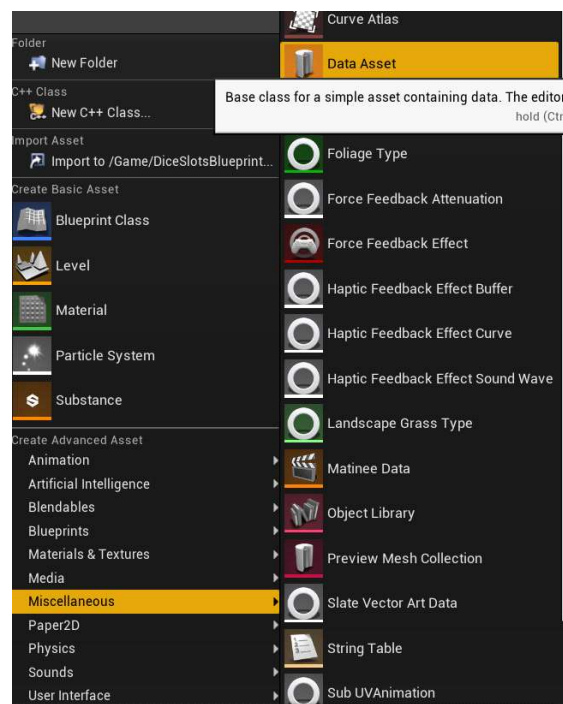
Finally the child widget's blueprint should look something like this (with the inputs hooked

up)



Creating Dice Slots Data Asset

First you need to create a data asset of type `DA_DiceSlotsVersion_DataAsset`.



Inside it you'll get to set some variables specific to the version of dice slots you're customizing.

Version Name: Name of the version of dice slots you're creating. This name is mostly used for saving data specific to the dice slots version.

WidgetClass: Basically the child widget blueprint you created in the last step.

2D Lobby Display Name: This name will be shown in the 2D lobby (which we'll get to setting it up later in the documentation)

2D Lobby Thumbnail: The background image used for this dice slots version to be shown inside the 2DLobby.

3CardsMultiplier: Points will be multiplied by this number in case players get points in 3 fields in the game.

4CardsMultiplier: Points will be multiplied by this number in case players get points in all 4 fields in the game.

BetOptions: It's just an array of integers. In the game players have the option to increase or decrease bet. Basically this array will be iterated to change the bet amount when they do want to change it.

Symbols: It's an array of type `F_Symbols_Structure`. Each symbol has 5 variables.

SymbolIdentifier: a unique identifier for each symbol. It has to be unique, otherwise 2 symbols having the same identifier will be treated as the same symbol.

SymbolChanceOfAppearance: a float between 0 and 1 which specifies how likely it is for that symbol to appear. Note that sum of your symbols chances of appearance should add up to exactly 1.

Points: amount of points that will be given to player in case they get this symbol lined up on a winning line.

IsBonuns: it's a Boolean to indicate if this symbol is treated as bonus or not. Basically lining up a bonus on winning line will not grant points, but it will grant mystery games to player.

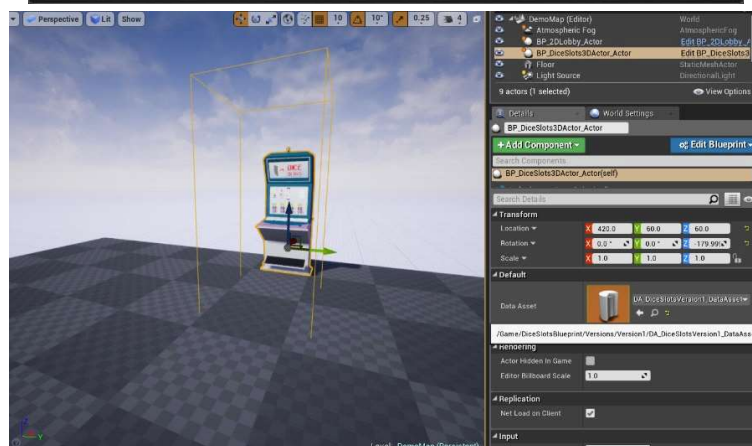
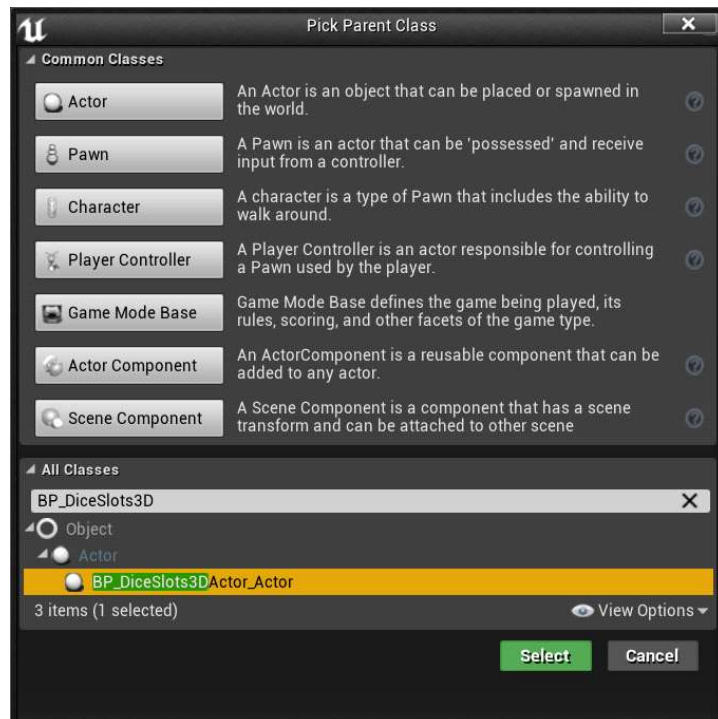
SymbolTexture: the texture that will be used for this symbol.

PointToReward: An array with each of its items being a structure containing a float and an integer. Basically it defines if player gets <Points> amount of points in the game, their balance should be added by <Reward>. Note that reward is not an absolute value and is a multiplier so in the end that reward is going to be multiplied by the bet player has placed.

DicePlacedSoundWave, LoseSoundWave, WinSoundWave: Just 3 sound waves and their purpose is pretty self-explanatory.

Setting Up a 3D Dice Slots Machine

You need to create a child of the actor called “BP_DiceSlots3DActor_Actor”. Inside it you can change the 3D model or its materials as you see fit. There is a DataAsset public variable attached to this actor and you have to set its value to the data asset you created in the last step. Place the actor in your scene and you’ll be good to go.



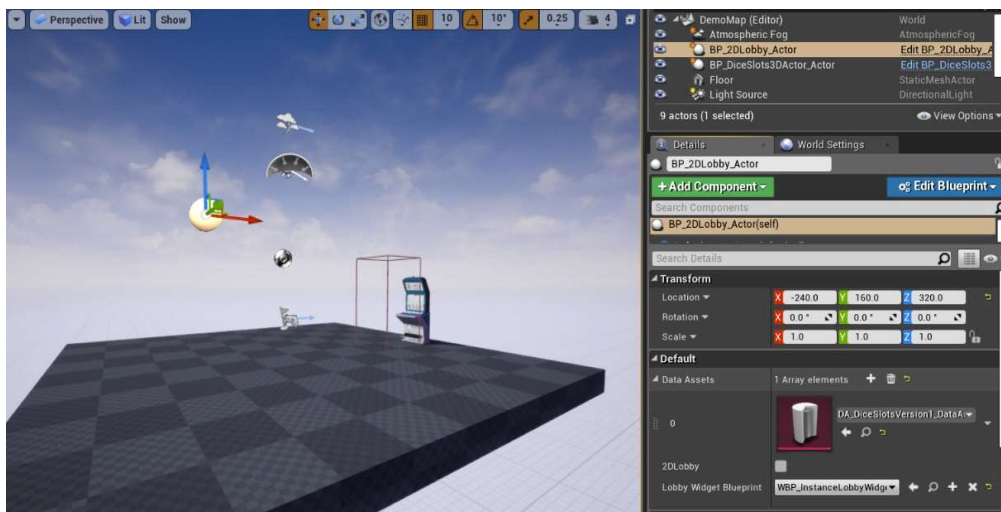
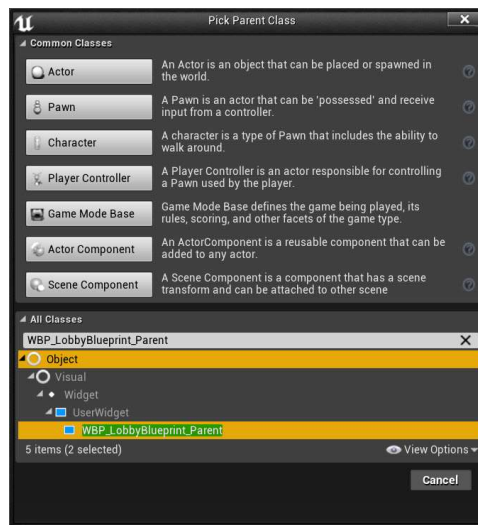
Setting Up a 2D Dice Slots

You can use Dice Slots child widget you created without using the lobby blueprint that exists in the project. All you gotta do is using the Create Widget node of unreal, set its widget to the child widget you created, then it will ask for the data asset as an input of the node, so you’ll need to feed it the data asset you created, then add it to viewport like you would do

with any other widget blueprint.

Setting Up a 2D Lobby

Basically you'll have to go through the same process as you did in the "Designing a new Dice Slots version" step, except instead of creating a child of "WBP_DiceSlotsMainGame_Widget" you need to create a child of "WBP_LobbyBlueprint_Parent", rename variables and redesign it. Next you'll need to add an instance of "BP_2DLobby_Actor" to your scene. It has 3 public variables on it.



2DLobby: A Boolean which you need to set to true if you want the 2D lobby to be shown as soon as scene is loaded.

Lobby Widget Blueprint: The child of "WBP_LobbyBlueprint_Parent" you created and customized.

Data Assets: An array containing all data assets related to all the Dice Slots versions you've

created and want to be shown in the 2D lobby.