Easy Dice Roller Pack

Documentation

Why:

There are a few dice packs available in the asset-store, but all of them had some critical issues. Our main goal was to make a dice pack where the dice will ALWAYS give a result when thrown. We achieved this by re-throwing the dice when they don't land properly, and even if the dice are continuously failing to land properly it will give a result.

We also wanted a variety of dice, easy to expand in future, and different dice for different players in the case of a multiplayer game.

How to use:

We tried to make the 'Easy Dice Roller Pack' as simple and organized as possible. In the prefabs folder you will find a 'Table' prefab. This Prefab will keep track of all the dice, how many need to be thrown, if they have landed, and what the result is. The Table contains the **Table.cs** component, where you can setup your table:

Result Counter: You can assign your own UI-Text field to it, which will show the results of all the dice.

Table Surface: This is the object that the dice will use to check the value they landed on (this object needs to have a collider component attached, and the dice will need to be placed above this objects so that they will fall on it)

SpinPower: how much spin the dice will have when they are thrown (randomly)

ForcePower: How much force the dice will get when they are thrown

ForceRandomDifference: How much variety there will be in the force that the dice will be thrown

ThowDirectionX,Y and Z: They direction that the dice will be thrown in (for instance setting X to -1 and Y and Z to 0 will throw the dice to the left)

The 'Table.cs' script also contains an integer list named **'ThrowResults'** where all results will be stored. We made this list public for easy access.

There are 2 ways to add dice to the Table.

NOTE: you can even combine both ways to add dice, which will result in certain dice being spawned and others being thrown from a fixed location.

1: Drag a die prefab into the Table prefab, making it a child of the table. By doing this, you can position the dice you want to throw exactly where you want them to start. The dice will add themselves to the table's list. The Table will be the one keeping track of the dice. They will be thrown in the direction and force you set in the component of the **Dice.cs** component that's attached to the die prefab. Look at **DiceExample2** for more information on how to get this up and running.

2: Add dice to the Table by script. You can access certain functions in the **Table.cs** script to add dice to the list that need to be thrown.

The functions are called:

AddDiceD4(int)

AddDiceD6(int)

AddDiceD8(int)

AddDiceD10(int)

AddDiceD12(int)

AddDiceD20(int)

By calling these function you will add the amount of dice that you specified with the integer you send with it. You can keep doing this as much as you want.

By calling the function **OnClearDiceList()** you can clear the entire list of dice that you want to throw.

By calling the function **OnResetDice()** you can clean up all the dice to throw them again.

By calling the function **OnThrowDice ()** the dice that added to the DiceToThrow list will be thrown.

The other important prefabs are the dice prefabs. They can be found in the Prefabs/Dice folder.

Each of these has a script called **Dice.cs** attached where you can change some settings:

Measure Distance: Is used to check if a side of the die landed on the surface. WARNING: Only change this if you change the scale of the dice. If this is set too high or too low, it can result in incorrect results.

SpinPower: how much spin the die will have when it is thrown (randomly)

ForcePower: How much force the die will get when it is thrown

ForceRandomDifference: How much variety there will be in the force that the die will be thrown

ThowDirectionX,Y and Z: They direction that the dice will be thrown in (for instance setting X to -1 and Y and Z to 0 will throw the dice to the left)

Rethrow Limit: How many times this die will be rerolled before it will be automatically calculated without physics. *(We made this for a worst case scenario when a dice keeps failing to land on the table.)*

Starting with a new scene:

You can find a prefab called 'MenuAndTable' inside the prefabs folder. By dragging this into your scene you are ready to add dice, throw dice and reset and clear the dice lists.

Starting from scratch:

-Drag the Table prefab into your scene.

-Create a UI button. Change its text to "add D6".

-Add an OnClick() list item to the button (by pressing +)

-Drag the Table prefab from your scene into this empty item

-Select the function Table.AddDiceD6 and set the amount to 1

-Create another UI button. Change its text to "throw dice".

-Add an OnClick() list item to the button (by pressing +)

-Drag the Table prefab into this empty item

-Select the function Table.OnThrowDice

-Press play and test the buttons.

NOTE: The Throw button will only trigger if all the results of the last throw are in.

-You can now expand by for instance adding a text field and dragging this text field into the Table.cs script that's attached to the Table prefab. (Select the Table prefab and drag the Textfield onto **ResultCounter**

Look at **DiceExample1** as an example.