

Dice Master - v1.1.0

Dice Master is a complete package for all your dice needs.
The system handles di?erent dice at runtime and allows you to create custom dice from a few textures.
The system is designed to be easy to use, so you can start rolling as soon as possible.
Common dice examples are also already set-up for you and ready to be used.

Quick Start

This section will showcase the example scenes included with the tool.

Common concepts

Dice Master works through a set of components that can be attached to GameObjects. Each component has a specific function, as follows:

- **Dice:** this component is attached to each dice and defines its behaviour, as well as exposing an event to catch dice throw results.
- **Thrower:** this component will apply a linear force to the dice it is attached to.
- **Spinner:** this component will apply a rotational force to the dice it is attached to.
- **Spawner:** this component instantiates dice prefabs in an arrangement. This can be combined with a Thrower and a Spinner to throw / spin the dice.
- **Roller:** this example component is a simple interface that allows the user to control when a Thrower and Spinner are triggered through key presses.

Note: the green and pink lines that appear when you attach a Spinner or a Thrower will help you visualize the dice trajectory when spun/thrown.

Example scene: ExampleDiceAll

This scene showcases all the dice that are already included with the tool.
These dice work as examples of what can be done with the tool, although they are ready to be used for your games.

Example scene: ExampleDiceMassSpawn

This scene showcases mass spawning of dice.
The scene is setup with a Spawner that will create a set of D6 when the user presses the 'S' key.
The Spawner also has a Thrower and Spinner attached, which will make the newly created dice jump and rool at startup.

Example scene: ExampleDiceResultRegister

This scene showcases dice result registering, i.e. checking the results of dice.
The scene is setup with 3 D6 which are spinned as the scene start (notice the Spinner attached to the 3 dice GameObjects).
The TestDiceRegister script registers to the dice events, so that when the dice is resting the register will get all the results and add them.

Example scene: ExampleDiceRoller

This scene showcases dice rolling.
Notice that the provided D20 has a Thrower and a Spinner.
These two components are triggered through the Roller script.
The ResultUI canvas element registers to the D20 dice results and shows them to screen each time it is rolled.

Example scene: ExampleDiceSpawnSelector

This scene showcases a mix of dice spawning and result registering.
The SpawnSelector script will use a Spawner to spawn di?erent dice as selected from the in-game UI.
The TestSpawnRegister will attach to the Spawner and wait for a dice to be spawned, so that it can collect the result.

Example scene: ExampleDiceSpawnFormat

This scene showcases spawning dice using a dice string, in the format of "5D6".
The SpawnFormat script will use a Spawner to spawn di?erent dice as selected from the in-game UI.

Example scene: ExampleDiceDynamicFaces

This scene showcases a dice with dynamic faces. These faces are textured quads, instead of being drawn on the texture, which allows runtime code to edit and animate the faces.
The TestCustomFaces will attach to the Dice and animate the shown face when it is read.

Example scene: ExampleDiceCustomFaces

This scene showcases a dice with custom faces. These faces could be any gameobject and can be accessed from the Dice itself.
The TestCustomFaces will attach to the Dice and animate the shown face when it is read.

Basic Usage

This section will teach you how to setup Dice Master for common uses.

How to: Roll an existing dice

- Choose your dice prefab from the DiceMaster/ExampleDice/Prefabs folder.
- Place the dice you chose inside your scene.
- Attach a Thrower and/or a Spinner to the dice and edit its properties.
- Add some form of platform the dice can rest on (such as the table prefab you can find in DiceMaster/Assets/Prefabs/Table)
- Press play!

How to: Detect the result of a dice throw

- Select the dice you want to check in your scene
- In the Inspector, add a new event listener to the OnShowNumber event GUI.
- Select a GameObject from the dropdown menu of the event GUI, making sure it has a script that can handle the result (for an example, see the ResultUI script, which you can attach to any uGUI Text gameobject).
- Set the event target to a dynamic function that takes an integer as an input (such as the HandleResult function in the ResultUI script)

Note: you can check the TestDiceRoller example scene to see the result.

How to: Spawn a dice

- Create a new GameObject in your scene and attach a Spawner to it.
- Drag the dice to be spawned in the 'dicePrefab' variable of the Spawner.
- Optionally, attach a Thrower and/or Spinner to the Spawner GameObject (make sure they do not have TriggerOnStart enabled).
- Trigger the Spawner (see the TestDiceSpawner script for an example).

Note: you can combine spawning and result checking! See the TestSpawnRegister script for an example of use.

Creating new dice

If the default dice are not enough for your needs, you can use Dice Master to create new dice prefabs.

You may need the following assets (although common assets are already provided for you to use!):

- A DiceDefinition custom asset. This defines how many faces the dice has and how they are arranged. Defaults for all common dice are provided, but you can add your own if you need.
- A dice 3D mesh. You can use one of the provided default meshes which cover all common use cases!
- A pattern texture, which will be used as the background color of the dice. You can use one of the default patterns or add your own. Make sure these textures are read/write enabled.
- A set of pip textures, which will be placed on the faces of the spawned dice. You can use one of the several default pips provided or add your own. Make sure these textures are read/write enabled.
- A material. I suggest to use the default PBR material which gives great results.
- Optionally, a physical material. The default physical material simulated plastic dice.

Once you have everything, proceed as follows:

- Go to Window->Dice Creator. The DiceCreator window will appear.
- Drag a DiceDefinition from the DiceMaster/DiceDefinitions folder to the window. The DiceDefinition will define how a dice behaves (like a D2, a D4, and so on).
- Give a name to your new dice.
- Choose a pattern texture, a mesh, a collision mesh, a material and a physical material. Defaults are provided.
- If you want to change the pips (the face textures), toggle "Show faces options" on and drag the new face textures in. You can also edit their relative size and orientation.
- Press Generate Prefab to create the final dice, you will find it (by default) in the DiceMaster/_Output/Prefabs folder!

Note that pips/faces have various parameters you can edit to change their appearance and arrangement on the faces:

- **Orientation** applies a rotation to the face. The given default value is provided by the DiceDefinition and mimicks real dice.
- **Size** scales the face texture. Make sure not to make this too large or the texture could overflow the face!
- **Stretch** scales the face texture vertically. This can be changed to achieve better results.

Accessing the API

All APIs are located in the `DiceMaster` namespace.

System Configuration

You can configure the system by editing values in the DiceConfig custom asset instance, which can be found in the DiceMaster/Resources folder. Note: Due to a bug with the Unity Asset Store, paths are not exposed in the inspector and must thus be edited by editing the DiceConfig.cs file directly. Here is a rundown of what you can edit:

- Parallel Tolerance: this value defines the threshold used for checking face orientations. The smaller this is, the more accurate the face computation will be.
- Check Speed Threshold: this value is used to determine below what speed we will check what number has been rolled. A higher value will make the check faster but less accurate.
- Overriden Max Angular Velocity: enlarge this if your dice roll at a slow motion. WARNING: DO NOT GO ABOVE 100 OR THE PHYSICS SIMULATION WILL BEWRONG.
- Project Path: this is the path of DiceMaster in your project. Change this if you are moving your folders around.
- Output Path: this is the path of output prefabs, textures, and materials, generated by the Dice Creator. This is relative to the project path.
- Default Pips Path: this is the path where Dice Master expects to find the default pips textures.
- Default Meshes Path: this is the path where Dice Master expects to find the default meshes.
- Default Materials Path: this is the path where Dice Master expects to find the default materials.

FAQ

Troubleshooting

- **Q - The dice seem to fall in slow-motion**

A - A Unity unit is equal to 1 meter, so the dice by default are bigger than in reality! This is easily solved by either (I) increasing the gravity in the Physics settings (30 looks like a good value) or (II) making the dice smaller. I suggest using option I to avoid issues with collisions.

- **Q - When switching to Android and creating a new dice prefab, I got an error about texture format.**

A - If this happens, make sure that all the input sprites (pattern and pips) are set to have RGBA32 compression

- **Q - My dice always show the same number, if I restart the scene multiple times.**

A - You can use the Random O?set value in a Thrower or Spinner to apply a random di?erent initial condition which will change the dice's result.

- **Q - How can I stop dice from rolling outside of the camera view?**

A - You will find a Prefab named Limits, which contains collision planes you can use for the purpose. Move them around as needed.

- **Q - Can I use a di?erent gravity vector?**

A - Yes, the system will adapt to the current world's gravity, as chosen in the Physics settings.

- **Q - Dice's rotations are not realistic, they are slow!**

A - Unity will limit the maximum angular velocity of all rigidbodies. You can edit the max angular velocity value in the DiceConfig custom asset to change the default behaviour.

Dice Creation

- **Q - Where do I find the dice created with the Dice Creator?**

A - By default, you will find the generated dice prefabs in the DiceMaster/_Output/Prefabs folder. You can change that by editing the DiceConfig asset.

- **Q - Can I use my own pips / faces?**

A - Yes, and it's easy! Just import your pip textures as sprites in the texture import settings, and make sure you make them read/write enabled in the advanced settings. You can then use these as pips in the Dice Creator.

- **Q - Can I use my own textures as patterns?**

A - Of course you can! Be sure to set them as read/write enabled in the Advanced settings of the texture importer. Also, keep in mind that the UVs are specifically made so to make texture generation possible, so seams may be seen. You can check the UV layouts of all dice in the "DiceMaster/GFX /Textures/UV layouts" folder and draw upon them, if you prefer.

- **Q - Can I use my own meshes?**

A - You can, but you should mimic the UV layouts that are provided if you want Pips to be placed correctly.

- **Q - Can I create a new type of dice if I have its mesh?**

A - Yes, you can. You can create your own DiceDefinition class that will define how the new dice behaves. If you already have textures, then defining the directions for each face and their values will be enough. If you want to support Pip creation, you need to create a correct UV layout and specify the UV center of each face.

- **Q - Creation seems to get stuck loading.**

A - Larger textures may slow down the texture generation part. Try reducing the target texture size in the Dice Creator window, or using smaller pip textures.

- **Q - My rotated pips are cut on one side!**

A - That may happen because the input pip texture is too large to be rotated in-place. To fix that, set the Rotation Texture Multiplier value in the DiceConfig to something higher than 1 (1.4 should do). Note that this will increase texture generation time.

- **Q - My pips are skewed! What happened?**

A - For best results, make sure your pip textures are power of 2 and square.

- **Q - How can I *rig* the dice?**

A - If you want to rig your dice, you will need to define weights for all the faces. Weights are defined in the DiceDefinition. Just copy the DiceDefinition you need and adjust the weights (see the weighted_D6 DiceDefinition for an example).

- **Q - How can I use dynamic or custom faces?**

A - In the Dice Creator window, check Show Face Options, then select the desired FacesType from the dropdown menu. Dynamic faces are textured quads, while custom faces can be any GameObject.

Contacts

For any enquiry, for support, and for feature suggestion, [contact me](#)