Deterministic Dice Roller Documentation

Contents

Supported Versions

Tested and supported for Unity version:

2018.3

2019.1

2019.2

2019.4

2020.3 LTS

Unity 6

Release Notes

1.0

Initial release.

1.1

Fixed a bug where if the dice landed below x=0 it had some issues finding the top roll.

Added “Set Kinematic” which sets all other rigidbodies to “set Kinematic” during the Simulation and unsets them afterwards.

1.2

SkullDice added.

1.3

Updated for 2020.3

Removed physics objects that could cause dices to incorrectly show the right number.

1.4

Added D8 and D20 dices.
Added 25 new handpainted textures.

Added 3 new Physics handling modes:
Set Kinematic.
Collide with Rigidbodies.
Collide with Rigidbodies and Freeze.

These physics handling modes are for when there are lot of rigidbodies in the scene the dices can collide with, provides better handling for those scenarios.

Performance enchancements.
Cleanup in code.

Added another skulldice (does not work to roll deterministically.)

1.5

Added support for Unity 2D mode (Gravity is Z 9.81).

Added D4, D10, D12, D16 dices (And 32 textures for the dices)

1.6

Added truncated D4.

Exposed the direction Vector so you can choose the direction you want to throw the dices without making changes to the code.

Added physics bounciness material for those who wants more bounciness.

Cleaned some code.

1.7

Refactor

1.8

Support for unity 6

FAQ/Troubleshooting

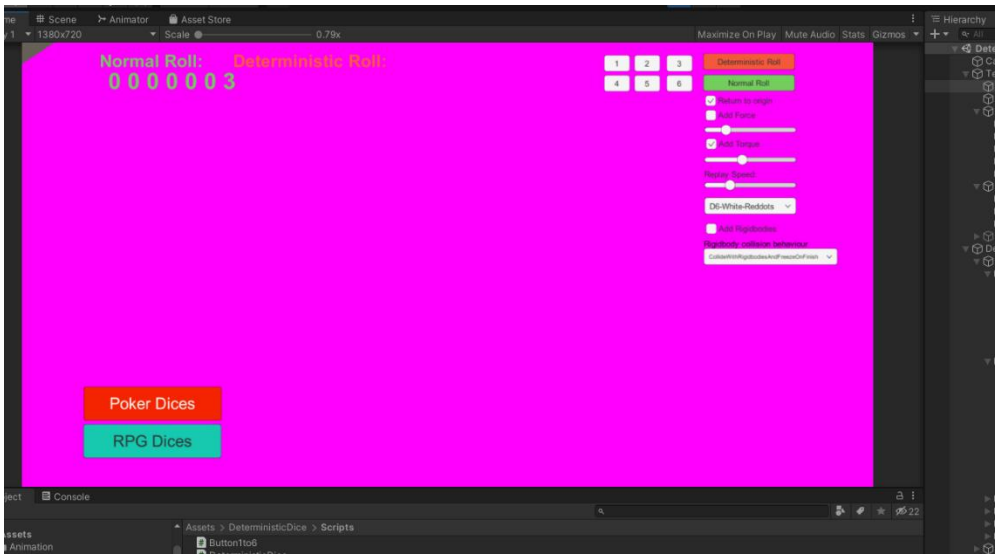
- Why is there a difference in speed when doing a normal roll and a deterministic roll?

A normal roll uses the Gravity settings of unity, while a deterministic roll uses the replay speed setting. Change the replay speed setting/or gravity settings to suit your needs.

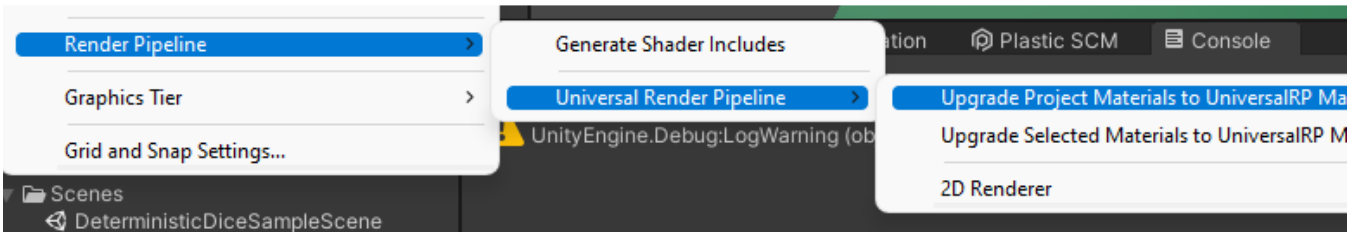
- I receive Warning “WARNING: Dice D6Deterministic (X) (UnityEngine.GameObject) did not finish the simulation! Most likely it fell of the board/disappeared. Did you add to much force to it?” in the logs?

This happens when the dices are not able to finish the simulations within the default step value (Default is 2000 steps). This usually happens when you don’t have any colliders on the plane, and they fall off. Make sure all dices come to rest. Preferably as quickly as possible.

- All assets are purple when opening the package!



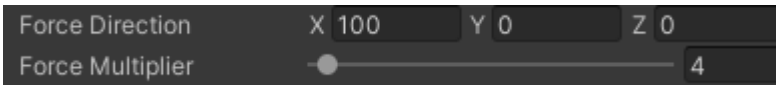
If you use URP or HDRP you need to convert the materials to URP/HDRP:



- How can we achieve different initial trajectories for the dice? Like throw them downwards or left or right etc.

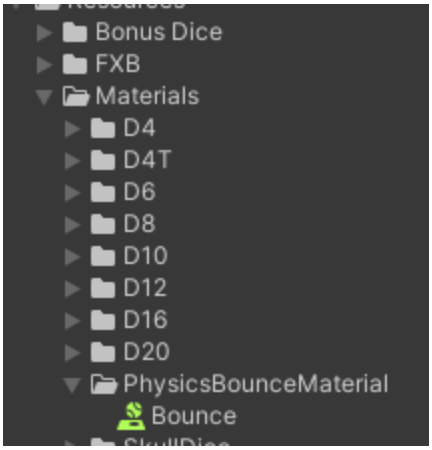
  By setting the Force Direction:

  Example: X=100 will throw the dices from left to right

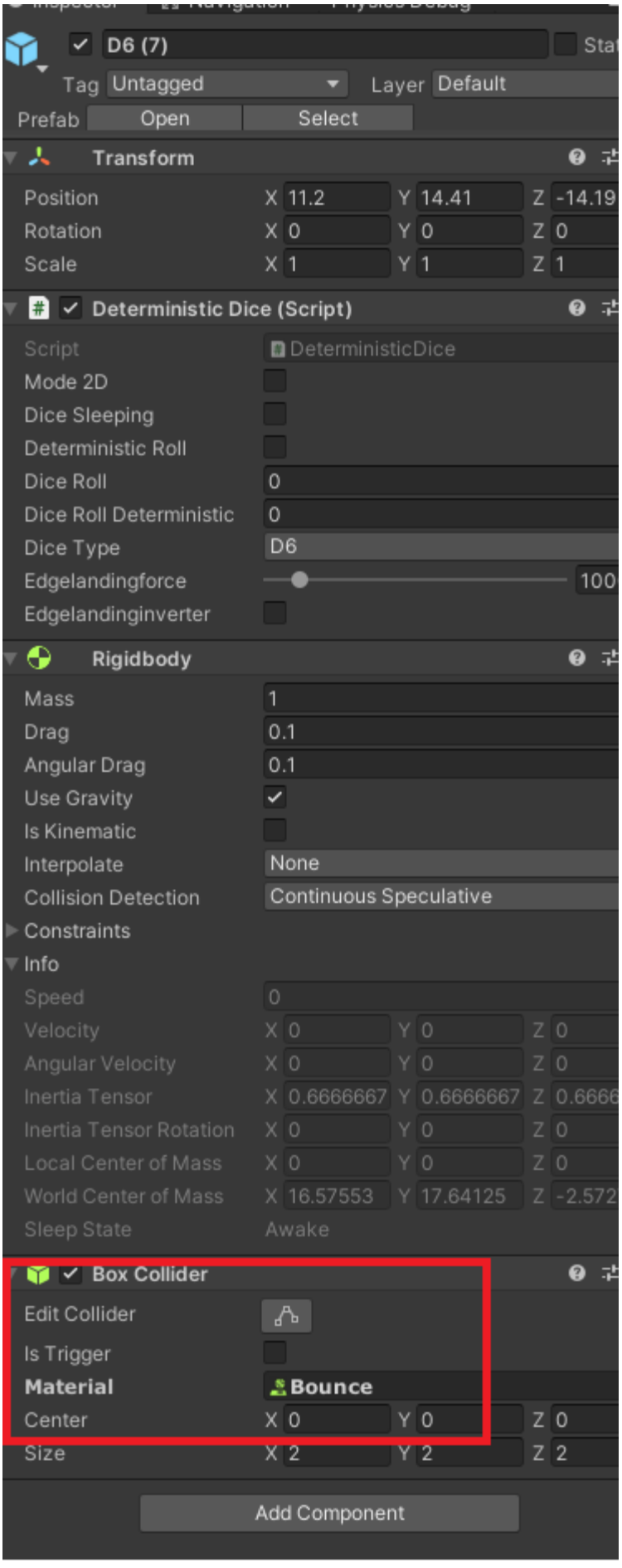


- Is there a way to increase the “bounciness” of the dices?

  A physics bounciness material is included:



  Add this to the dices and it will increase bounciness:

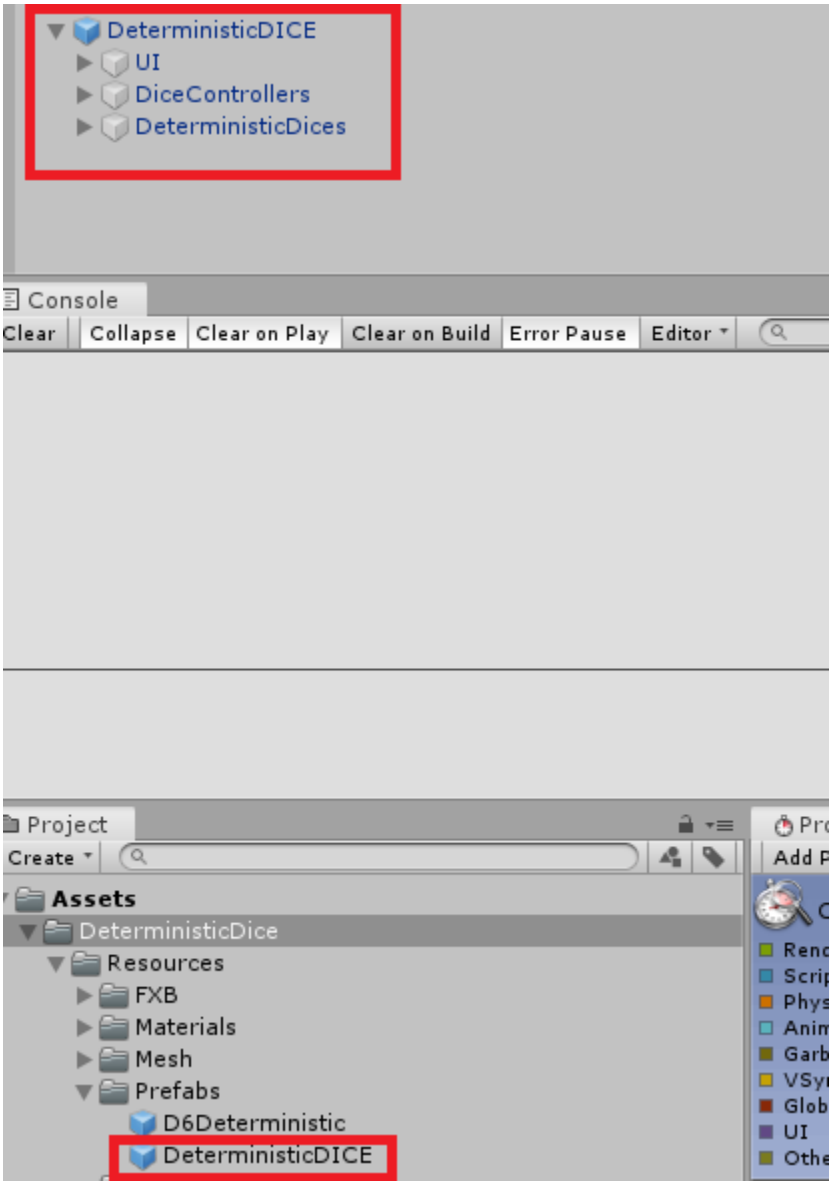

D6 (7)
Tag Untagged
Layer Default
Prefab Open Select
Transform
Position X 11.2 Y 14.41 Z -14.19
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Deterministic Dice (Script)
Script DeterministicDice
Mode 2D
Dice Sleeping
Deterministic Roll
Dice Roll 0
Dice Roll Deterministic 0
Dice Type D6
Edgelandingforce 100
Edgelandinginverter
Rigidbody
Mass 1
Drag 0.1
Angular Drag 0.1
Use Gravity
Is Kinematic
Interpolate None
Collision Detection Continuous Speculative
Constraints
Info
Speed 0
Velocity X 0 Y 0 Z 0
Angular Velocity X 0 Y 0 Z 0
Inertia Tensor X 0.6666667 Y 0.6666667 Z 0.6666
Inertia Tensor Rotation X 0 Y 0 Z 0
Local Center of Mass X 0 Y 0 Z 0
World Center of Mass X 16.57553 Y 17.64125 Z -2.572
Sleep State Awake
Box Collider
Edit Collider
Is Trigger
Material Bounce
Center X 0 Y 0 Z 0
Size X 2 Y 2 Z 2
Add Component

Settings for bounciness can be modified on the material

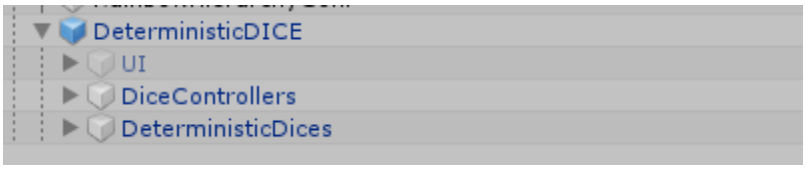
Quick install guide

The fastest way to get going with your project:

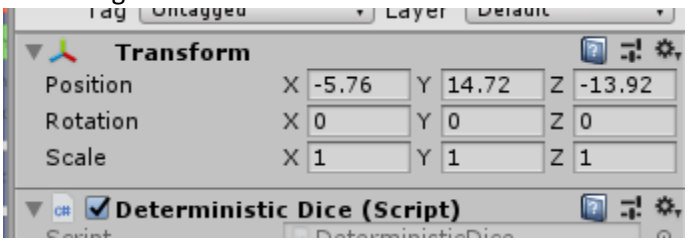
1. Import the package via the asset store
2. Drag the prefab under DeterministicDice->Resources->Prefabs->DeterministicDice to the Unity project hierarchy



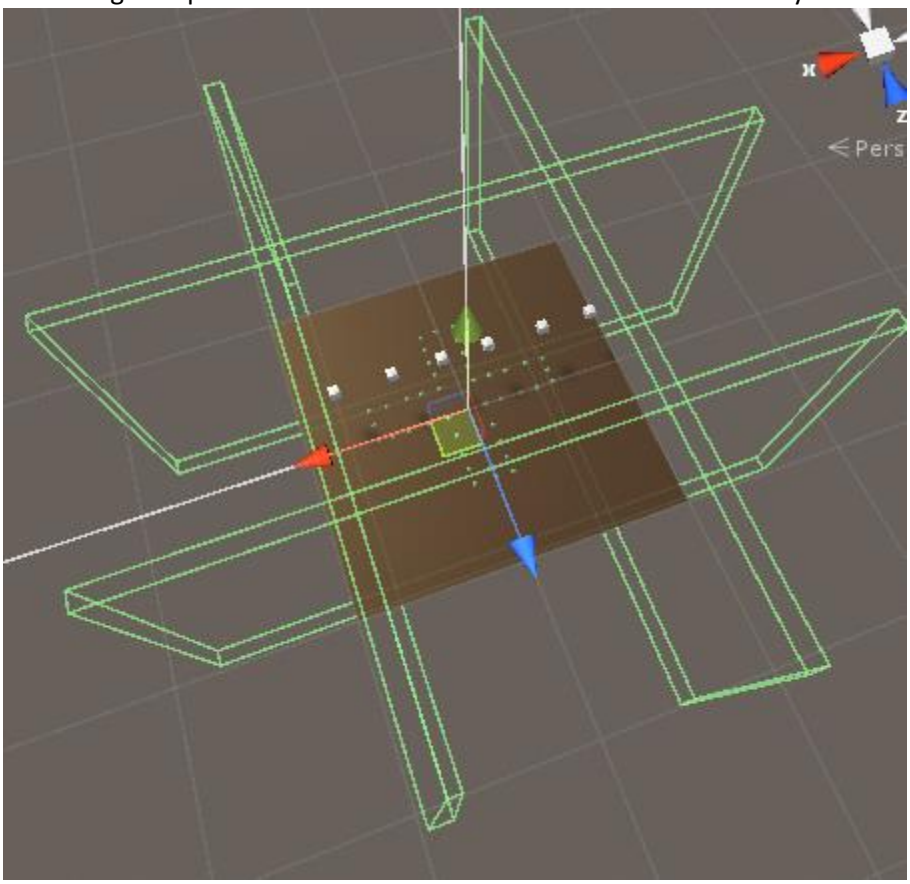
3. Disable UI component - The above will include the UI as well, this is normally not used so feel free to disable it. But it can be good for early troubleshooting/testing and setup of the Deterministic Dice roller.



4. Scale and correct the positions of the dices. They are all located under the game object Deterministic Dices. If you need to change the size of the dices open the D6Deterministic prefab and change that.

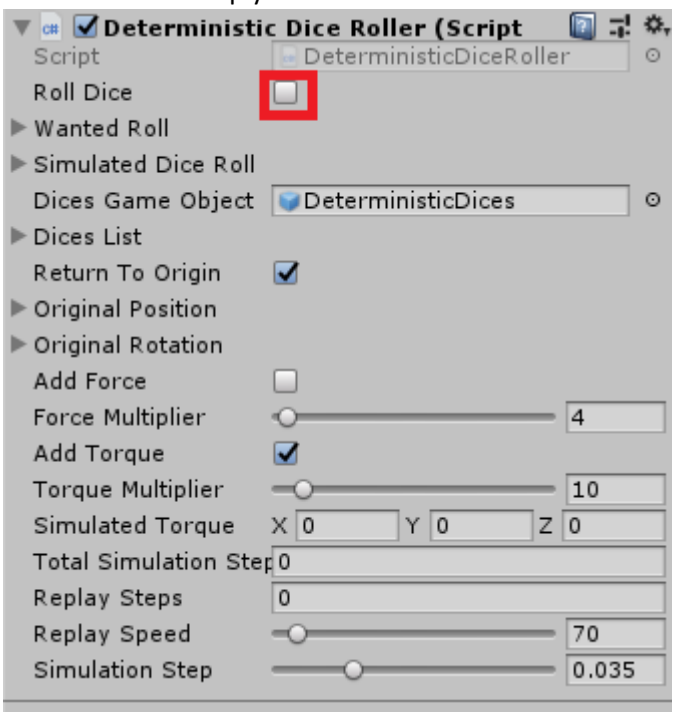


5. Don't forget to place colliders around the terrain to make sure they don't fall off.



6. Roll the dices!
   All the logic to roll the dices are contained in the DeterministicDiceRoller gameObject/script:

To roll the dice simply click the check box “Roll Dice”



To do this programmatically create a public variable and populate it with the diceRollScript:

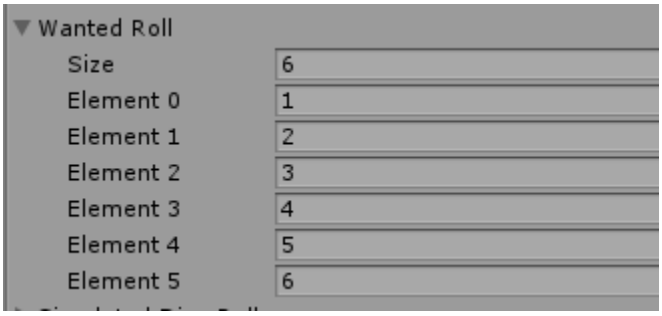
public DeterministicDiceRoller diceRollScript;

access the rollDice bool and set it to true:

```
// Start is called before the first frame update
void Start()
{
 //This is what rolls the dices
           diceRollScript.rollDice = true;

}
```

7. By default the dices will always roll a 1. This is true as long as you don’t populate the list in the DeterministicDiceRoller called “wantedRoll”.
   There needs to be one wantedRoll entry per dice under the Gameobject DeterministicDices. If you are rolling 6 dices, then there needs to be 6 entries:



To populate it simply add a number(int) to the list:

```
//Add number 5 to the list
diceRollScript.wantedRoll.Add(5);
```

Don't forget to clear the list between the rolls:

```
diceRollScript.wantedRoll = new List<int>();
```

8. For your convenience the UI script "buttons1to6" contains all kinds of rolling behavior and control. Please check it if you feel stuck.

Components

Components used:

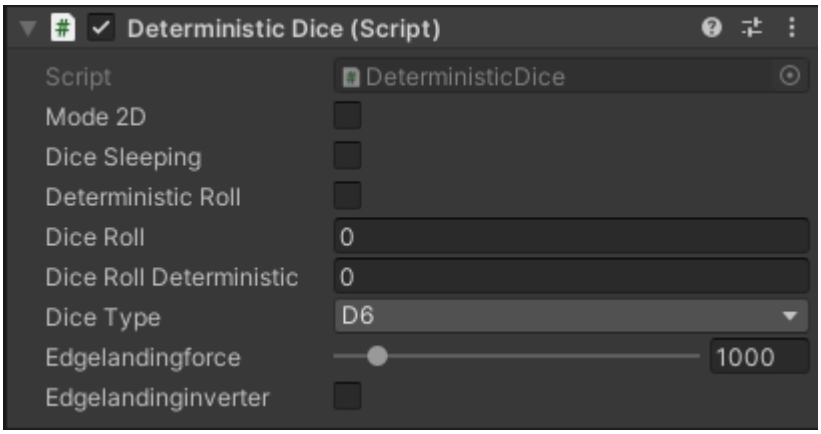
Name	Type	Description
UI	GameObject	UI contains all the buttons/sliders/toggles for testing. The idea is to get a feel for how it works, and then disable it.
ButtonController	GameObject	Runs the script "Button1to6"
DeterministicDiceRoller	GameObject	Runs the script "DeterministicDiceRoller"
DeterministicDices	GameObject	Holds all the dices. Every ACTIVE dice-GameObject under this GameObject will trigger a dice roll. So if you want to roll 3 dices, keep 3 dices active/enabled under this GameObject. You can either disable or delete dices to stop them from being rolled.
D6Deterministic	GameObject	The dice itself. Runs the script "DeterministicDice".

Button1to6	Script	Script for all the UI button logic. Check here if you need some inspiration on how to access the settings of DeterministicDiceRoller.
DeterministicDice	Script	The only thing this script does is to report which side is up when the dice has stopped rolling. Everything else is done in the DeterministicDiceRoller script.
DeterministicDiceRoller	Script	The main logic component of this asset. Contains all settings and actions.

Settings

Deterministic Dice Settings:

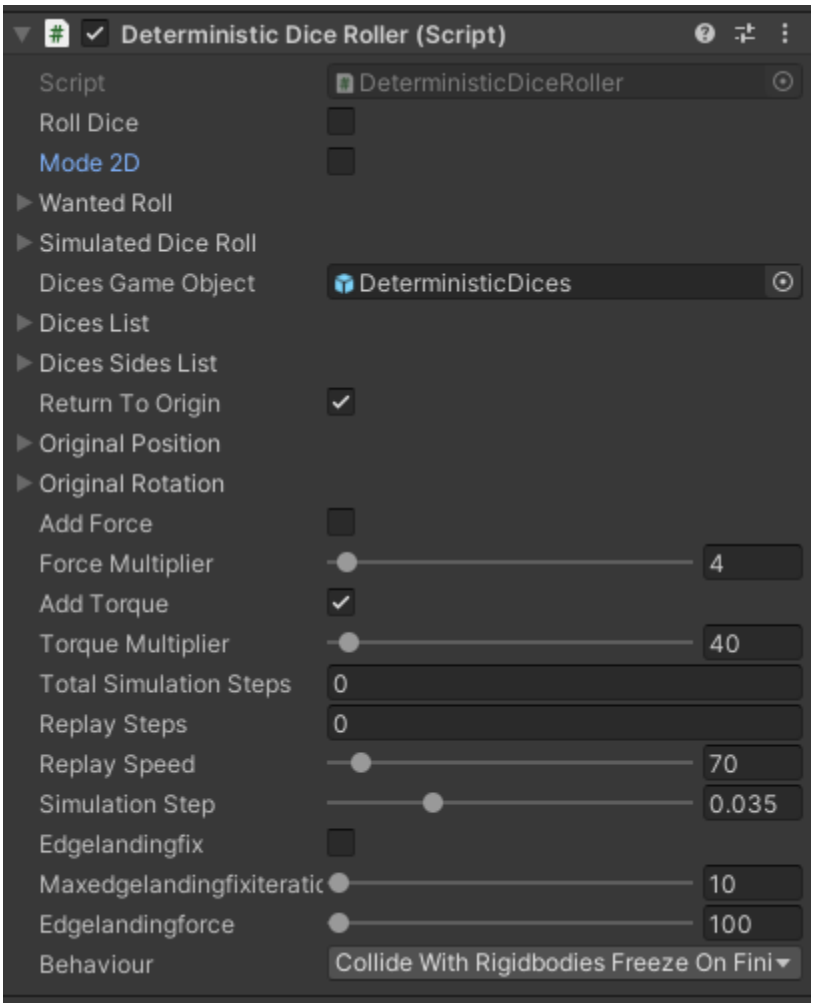
This script is attached to all dices and handles normal dice rolling as well as defines what kind of dice we have (D4,D6,D16 etc.)



Name	Type	Description
Mode 2D	Boolean	Set to true if you want the dices to work when Gravity is Z=9.81 (Unity Default 2D mode) when you are rolling them Normally.
Dice Sleeping	Boolean	Sets itself to true when the dices are resting, used to determine when the dices has stopped rolling.
Deterministic Roll	Boolean	Is set to true by the DeterministicDiceRoller script if we are rolling the dices deterministically
Dice Roll	Int	Gives the number the dice rolled if rolled normally
Dice Roll Deterministic	Int	Gives the number the dice rolled if rolled deterministically

Dice Type	Enumarator	Set to the dicetype. If wrong dicetype is chosen it will not work correctly. Supports: D4,D6,D10,D12,D16,D20
Edgelandinginverter	Boolean	Experimental code, do not use.

DeterministicDiceRoller Settings:



Name	Type	Description
Roll Dice	Bool	This is checked every Update() and if true rolls all dices under the GameObject DeterministicDices.
Mode 2D	Boolean	Set to true if you want the dices to work when Gravity is Z=9.81 (Unity Default 2D mode)

Wanted Roll	List<int>	A list containing the rolls we want to make. Defaults to 1 if not populated. Add 1 entry per dice. In the same order as diceslist.
Simulated Dice Roll	List<int>	A list containing the simulated rolls. Do not change or edit.
Dices Game Object	GameObject	The Gameobject that contains all the dices. Default DeterministicDices.
Dices List	List<GameObject>	A list containing all the dices to be rolled. It adds all Active dices under above GameObject. Do not edit this list, instead remove/disable/add/activate dices under the GameObject DeterministicDices.
Dices Sides List	List<int>	A list containing how many sides each dice has (eg D6, D8, D20)
Return to Origin	Bool	Set this is you want the dices to be moved back to their original position between rolls. If disabled dices will be rolled from the current position.
Original Position	List<Vector3>	If Return to Origin is true then we will move them to the position in this list. Dice1= entry1, Dice2= entry2 etc… You can edit this between rolls if you want to move them to some specific place before the roll. Normally though, its better to just move the dices themselves and leave Return to Origin as false.
Original Rotation	List<Vector3>	If Return to Origin is true then we will rotate them to the position in this list. Dice1= entry1, Dice2= entry2 etc… You can edit this between rolls if you want to rotate them to some specific rotation before the roll.
Add Force	Bool	Set to true if you want to add random force to the dices when rolling. This can be done manually instead.
Force Direction	Vector3	The direction you want to throw the dice, example: Set Y = 100 and the dices will be thrown upwards. Remember to enable Add Force when using this. The result will be cleared if you disable torque during testing.
Force Multiplier	Int	Governs how much force is added when rolling. force = force * forceMultiplier.
Add Torque	Bool	Set to true if you want to add random torque to the dices when rolling. This can be done manually instead.
Torque Multiplier	Int	Governs how much torque is added when rolling. torque =torque * Torque Multiplier.

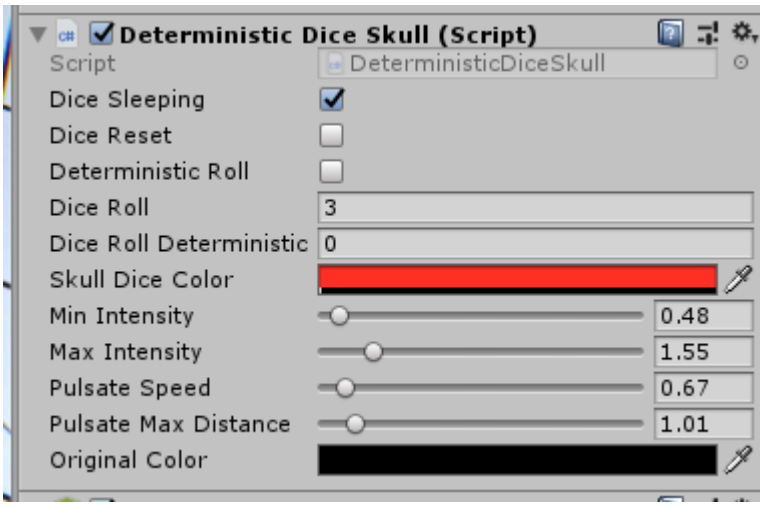
Simulated Torque	Vector3	Contains the applied randomized Torque.
Total Simulation Steps	Int	How many steps it took to simulated all dices until they stopped rolling (Maximum default is 2000). Normal is ~100-400.
Replay steps	Int	Where we are in the current replay. Starts at 0 and goes until we hit Total Simulation Steps.
Replay Speed	Int	How fast we replay the dices.
Simulation Step	Float	How many steps we will simulate and record. Default is to record every 0.035 step. Changing this effects performance. If you record less(e.g 0.10) you will get increased performance, where as 0.01 will simulate and record every step. If you want perfect slow motion you need to record at 0.01. Changing this also prompts changes to the replay speed.
Set Kinematic	Bool	sets all other rigidbodies to "set Kinematic" during the Simulation and unsets them afterwards. This decreases performance if there is a lot of rigibodies in the scene. If this is unset then all rigidbodies which is not asleep will move to their "final" destination.
Unset List	List<GameObject>	The list of all rigidbodies that will be set back to "use gravity".
Edgelandingfix	Bool	Experimental code to fix dices landing on edges, do not use.
Behaviour	List	Three physics behaviours have been added in 1.4: Set Kinematic. Collide with Rigidbodies. Collide with Rigidbodies and Freeze. These behaviours guide how objects will collide when simulating the rolls. If you have no other moving objects in the scene with which dices will collide with, use Set Kinematic for optimal performance. **Set Kinematic:** Use this if there should be minimum impact to performance and its highly unlikely other rigidbodies will collide with the dices. All Rigidbodies are set to kinematic during simulation, and unset afterwards. **Collide with Rigidbodies:**

		Other Rigidbodies are active during the simulation, uses a little bit more processing power but the dices should accurately collide with the rigidbodies and show the correct face up after colliding. **Collide with Rigidbodies and Freeze:** Other Rigidbodies are active during the simulation, uses a little bit more processing power but the dices should accurately collide with the rigidbodies After the dice has rolled, they will freeze in place. What can happen if this mode is not in use is that the simulation and replay is finished but there is still some movement going on that topples over one of the dices and then it will show the wrong number. With this mode the dice freezes and guarantees the correct number.

SkullDice

A skullDice has been added that changes color and pulses on the number it lands.

A few options to change color and speed/intensity of the pulse has been added:



Roadmap

- Fix edge landing
- Fix dynamically changing dices

- Check interpolation to save FPS
- Research doing multiple scene physics simulations