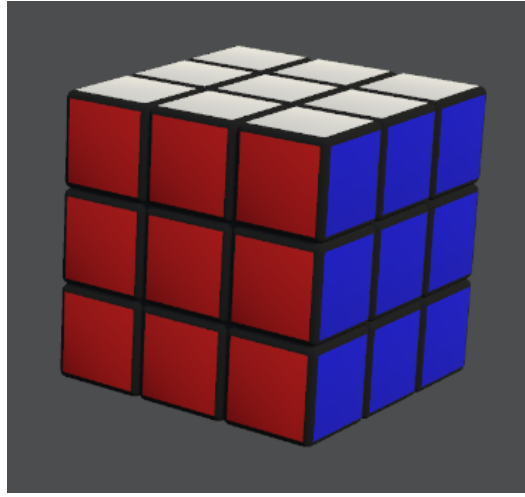


# Rubicks Cube Creator Manual



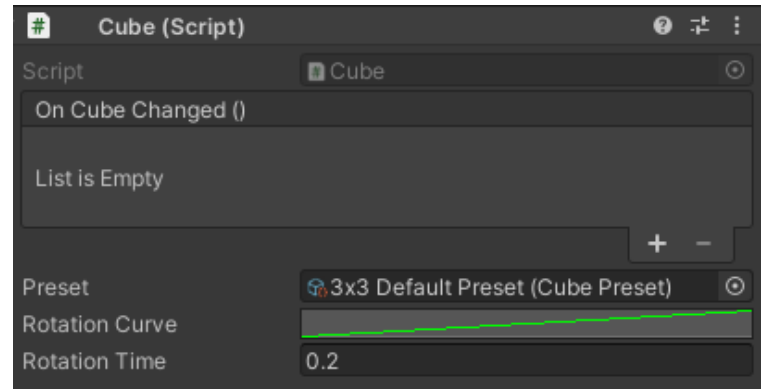
## Quick Start

To add rubick cube to your scene grab "Plugins/Rubicks Cube Creator/ Prefabs/ RubicksCube" prefab and drop it to your scene.

# Cube

## Inspector

- **On Cube Changed:** called every time cube plays a move.
- **Preset:** ScriptableObject that contains data for the cube such as layer count, block prefab and Materials.
- **Rotation Curve:** ease of the layer move.
- **Rotation Time:** time of the move.



## Script

- **bool CanMove** -> Tells if the cube can rotate (Setted false when moving)
- **CubeCore Core** -> Returns the core of the cube.
- **void CreateNewCube()** -> Creates new cube.
- **void UndoMove()** -> Undos last move played.
- **void PlayMove(CubeMove move)** -> Plays given move on the cube.
- **void Shuffle()** -> Shuffles the cube randomly (15 moves)
- **void Solve()** -> Solves the cube with the reverse of the played moves.
- **Block GetBlockOnThePoint(Vector3 point)** -> returns the block with a given point vector.
- **Color[] GetColorsOfFace(Sides side)** -> returns colors of the given side.
- **Sides GetSideWithNormal(Vector3 normal)** -> returns side of the cube with given normal vector.

# Cube Collider

**! Requires Cube Component and available preset on the attached object**

To add Cube Collider to the cube press Add Component on the Cube GameObject Inspector and press "RubicksCubeCreator -> Cube Collider"

This component automatically adds a box collider and sets the bounds relative to the preset dimensions count.

# Block

## Vector3Int StartPosition

Returns the start position of the block on the blocks grid.  
For example Left Bottom on 3x3 cube will return "0,0,0"

## Vector3Int CurrentPosition

Returns the current position of the block on the blocks grid.  
For example Left Bottom on a 3x3 cube will return "0,0,0" ignoring the start position.

## Vector3Int RealCurrentPosition

Returns the real current position of the block relative to the cube.  
For example Left Bottom on 3x3 cube will return "-1,-1,-1"

## Color GetVisibleSideColor(BlockSide side)

Returns the color of the given side if it's visible.

## BlockSide RelativeSide(Sides side)

Returns relative BlockSide with given cube side.  
For example blocks back is blue and facing towards left, calling this method with Sides.Left will return BlockSide.Back.

## BlockSide GetSideWithVector3(Vector3 direction)

Returns relative BlockSide with given vector direction.  
For example blocks back is blue and facing towards left, calling this method with Vector3.Left will return BlockSide.Back.

# Cube Move

## CubeMove(int layerIndex, CubeDirectionAxis direction, bool clockwise)

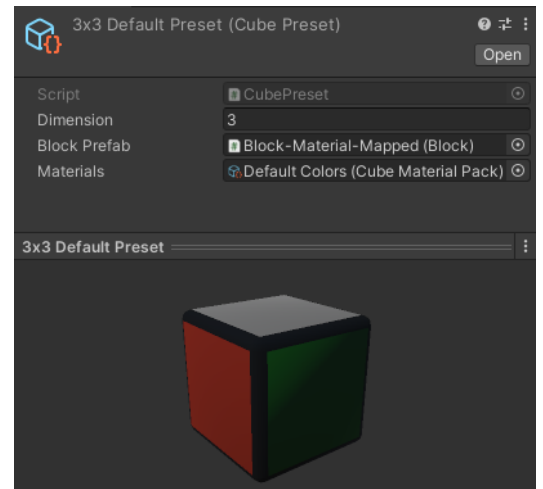
- **layerIndex**: index of the Blocks layer relative to direction.
- **direction** : Axis direction of the layer will rotate.
- **clockWise**: sets the direction of the move is towards clockwise

# Presets

## Cube Preset

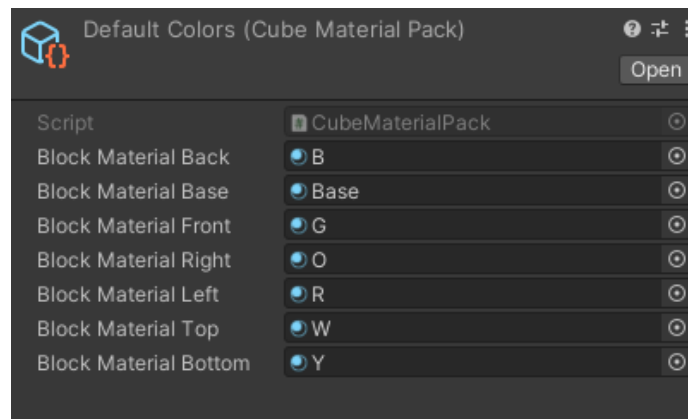
To create custom preset simply press right click to Project window, “Create-> Rubicks Cube Creator-> CubePreset”

- **Dimension:** Grid size of the cube.
- **Block Prefab:** Prefab of the blocks that are created. (If you want to change this prefab check the prefab and model located in the RubicksCubeCreator folder to create a new one).
- **Materials:** [Material Pack](#) for the RubicksCube.

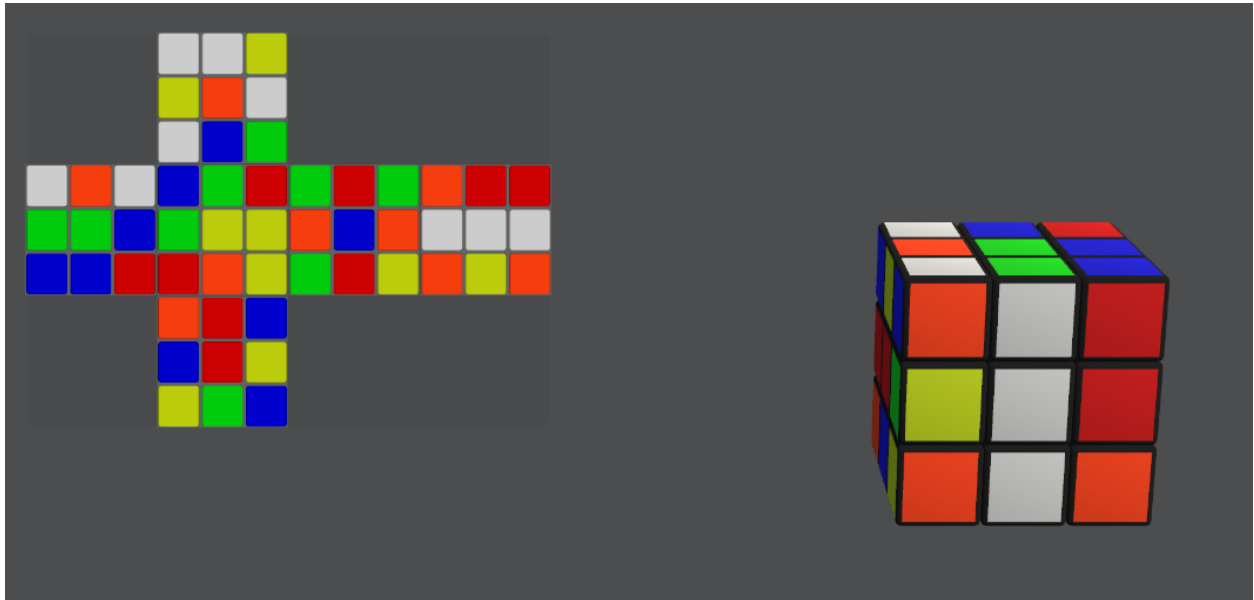


## Material Pack

To create custom material pack simply press right click to Project window, “Create-> Rubicks Cube Creator-> CubeMaterialPack”

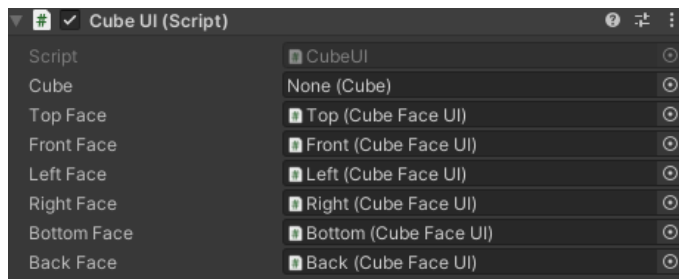


# UI



## CubeUI

Controller class of the face UI's ([CubeFaceUI](#))

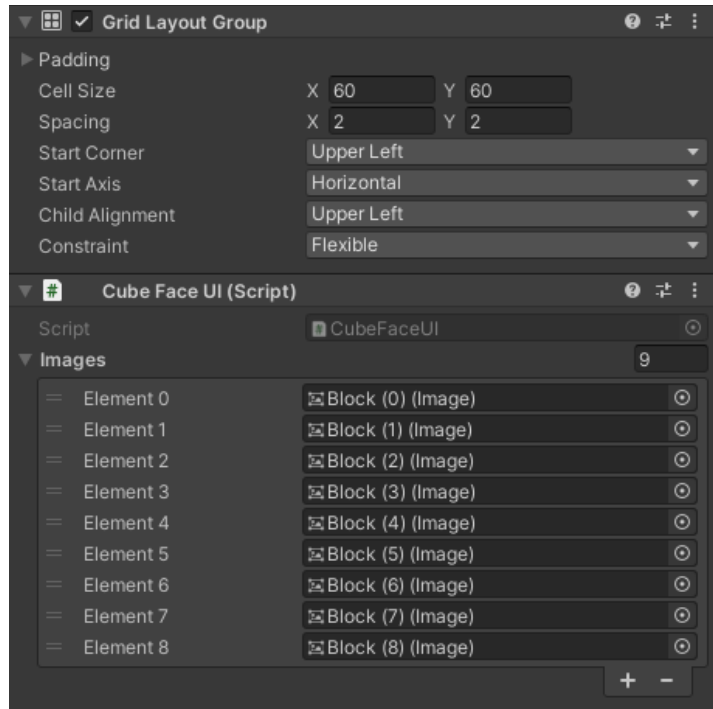


### How To Use

- Add CubeUI script to any object on the scene.
- Assign Cube to the script "Cube" variable.
- Setup [CubeFaceUI](#)'s for all sides of the cube.

CubeUI Automatically updates all the faces when a move is played by the cube.

# CubeFaceUI



## Setup

- Create UI Images that represent each block for example for 3x3 cube create 9 images on the UI.
- Attach all images to the CubeFaceUI component “Images” list.
- Assign the CubeFaceUI component to the [CubeUI](#)
- Additional : Use Unity’s Grid Layout Group component to set positions of the images like the cubes face.

You can check the Demo scene to see correct ways to implement the UI.

Asset pack already contains a 3x3 cube UI located in the Prefabs folder, check the prefab for further implementations.