Smart Rooms

Procedural Level Generator & Room Builder

USER GUIDE

Release 1.0.0

16. 7. 2023

Getting Started

Overview

This User Guide is crafted to equip ***Smart Rooms*** users with a basic understanding of the tool's features.

Setting Up

Upon downloading Smart Rooms from Unity's Asset Store, navigate to: "Assets->Import Package->Custom Package...". Locate and select the TicTacToeProSmartAI2048Minigame.unitypackage file in the Import Asset window. Once the "Importing package" window pops up in Unity, ensure all items to import are chosen and then click the Import button positioned at the lower right corner of the window.

Getting Started

A scene is prepared which contains both Tic Tac Toe and 2048 MiniGame. You can find the scene at TicTacToePro/Scenes/ShowCase.unity.
When you open the scene hit play in the Editor and play with the game a bit to explore it. Getting a good understanding of how the application works before diving deeper is crucial for building deep knowledge.

Help & API Documentation

For any inquiries or support needs, please visit our Discord channel at https://discord.gg/BqjyfzeYYX where you can access additional details, ask questions, get Video Tutorials, and FAQs. If you can't find the information you're looking for, feel free to ask for help in our channel.

Generators

SmartLevelGenerator

The SmartLevelGenerator class is responsible for generating a level based on a **LevelStyle**.

Settings

maxBuildIterations: An integer constant that represents the maximum number of build iterations allowed.

_levelStyle: A LevelStyle object that represents the style of the level to be generated.

_roomGameObjectsHolder: A Transform object that represents the holder for the room game objects.

_tilemap: A Tilemap object that represents the tilemap for the level.

_start: A Transform object that represents the start position for the level.

_exit: A Transform object that represents the exit position for the level.

_borderGenerator: A BorderGenerator object that represents the border generator for the level.

_backgroundGenerators: A list of BackgroundGenerator objects that represents the background generators for the level.

_sameTileNeighbor: A boolean value that determines whether or not the same tile can be used for neighboring tiles.

_randomStopChance: An integer value that represents the chance of randomly stopping the generation process.

_seed: An integer value that represents the seed for the random number generator.

_logGenerationTime: A boolean value that determines whether or not to log the generation time.

_logWarnings: A boolean value that determines whether or not to log warnings.

_maxAttempts: An integer value that represents the maximum number of attempts to generate the level.

_tryToFindError: A boolean value that determines whether or not to try to find an error during generation.

Methods

GenerateLevel(): A method that generates the level based on the LevelStyle.

GenerateBackground(): A method that generates the background for the level.

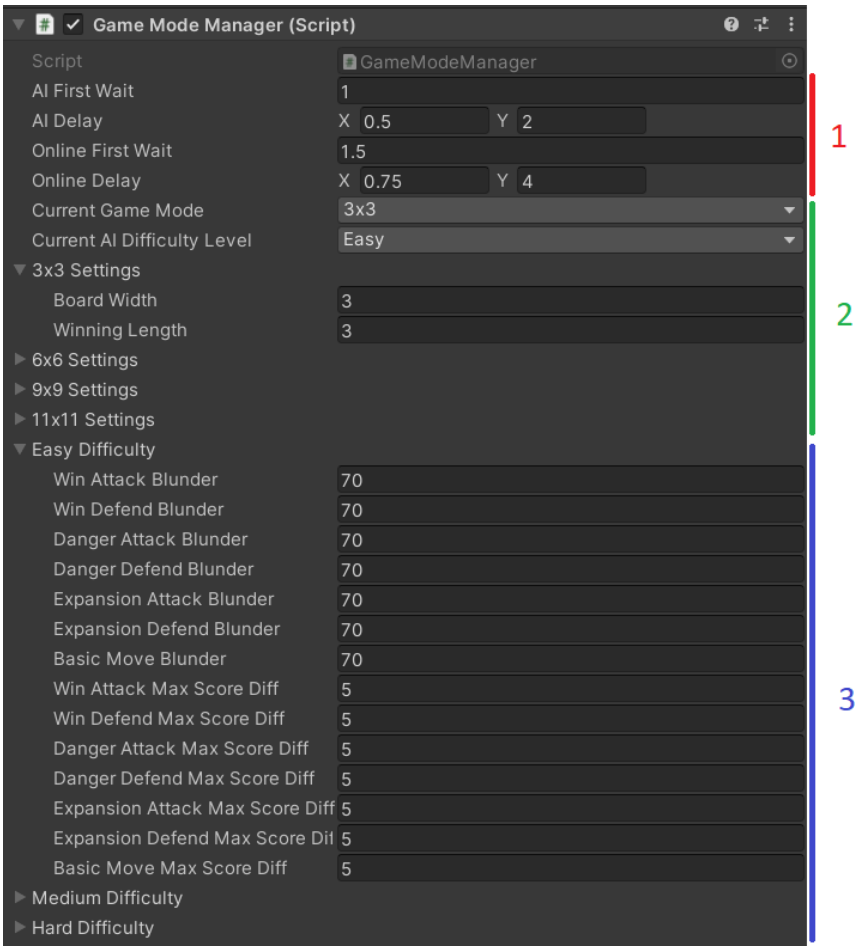
GenerateBorder(): A method that generates the border for the level.

Managers overview

Game Mode Manager

The Game Manager controls how the game of Tic Tac Toe is played. The manager is split into three categories of settings.

1) The gameplay settings for how long the AI waits before playing. It gives the player a more organic feeling than instant moves from the AI.
2) Settings for the different game modes. Don't touch these unless you want to change how the game is played.
3) Difficulty settings for the AI. The AI has a preprogrammed list of good moves it can do. If it executes the moves perfectly then it is virtually unbeatable. To allow players to win, we introduce the concept of Blunder and Max Score Diff.



Blunder: When the AI is doing a move it can Blunder it and go to the next possible move it can do, all the way down to Random move.

Max Score Diff: The AI can do multiple actions per specific move. To decide which action to take it calculates the score for each action and randomly chooses. The Max Score Diff tells which Actions can be used in the random search by comparing their score to the maximum score found for that move.

Screen Manager

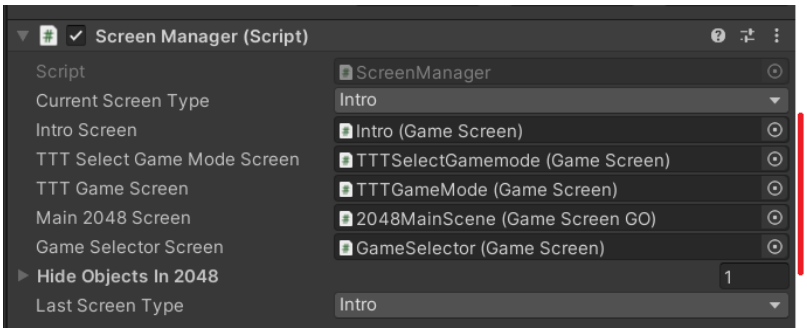
To make switching between screens seamless, this game utilizes the Canvas Group component. By animating the alpha and disabling the Canvas Groups it gives smooth transitions between screens and reduces any garbage collection related stuttering. The Screen Manager is handling the switching of each screen.

Current Screen Type: This tells which Screen is currently displayed. It also tells which Screen will be shown first in Build.

1) These are the main screens which can be switched between.

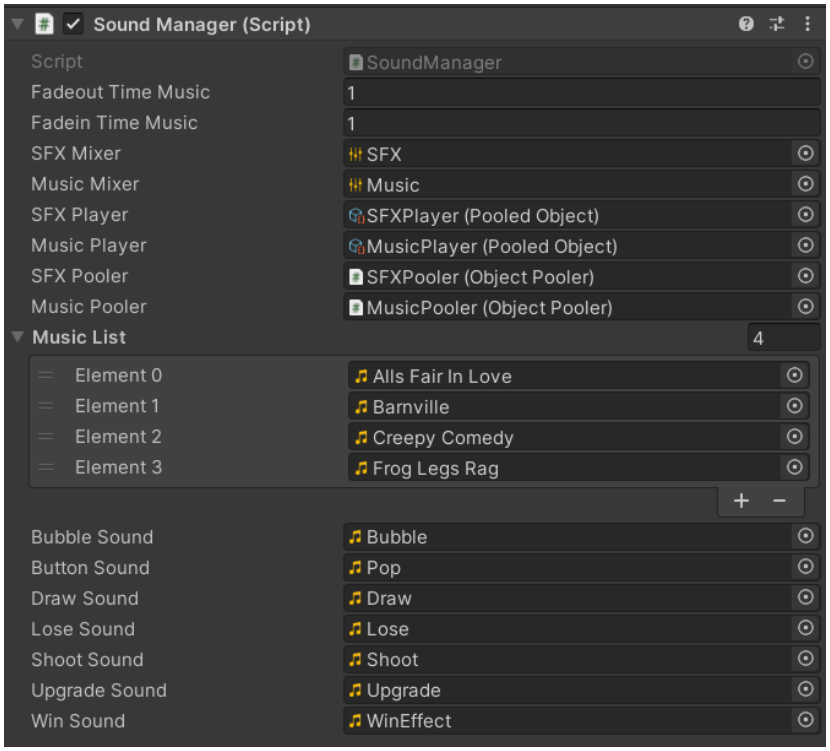
Hide Objects In 2048: We have to hide the canvas background which is used in Tic Tac Toe.

Last Screen Type: Used when the user needs to go Back.



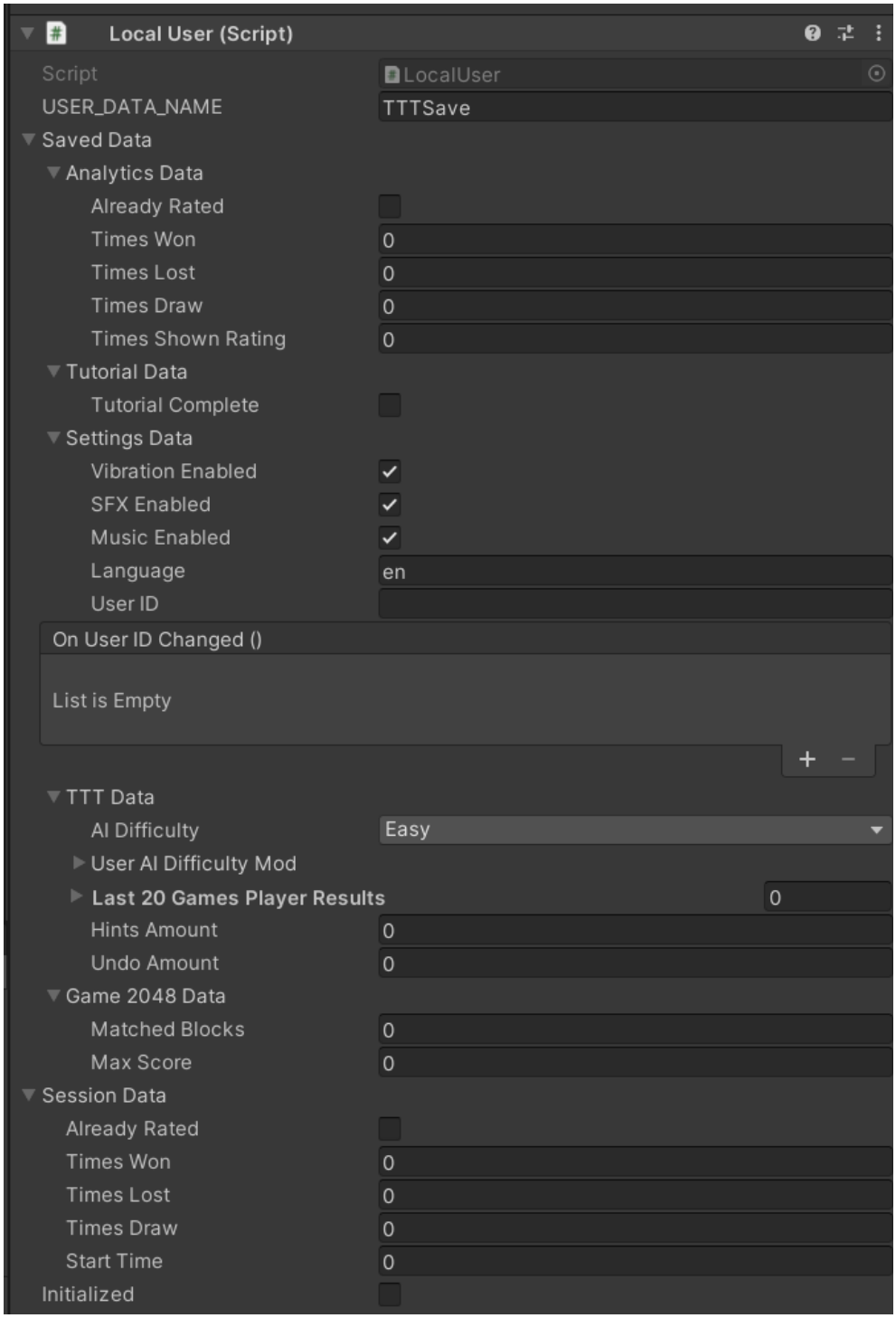
Sound Manager

Sound Manager is responsible for playing SFX and Music. It can be prompted from any script to play one of the Serialized SFX sounds. It serializes all of the clips needed to play in the game, so the SFX and Music is played regardless of the lifetime of any other object. It uses object pooling to play SFX in parallel.



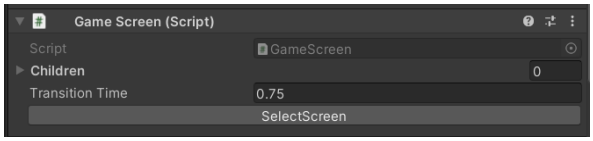
Local User

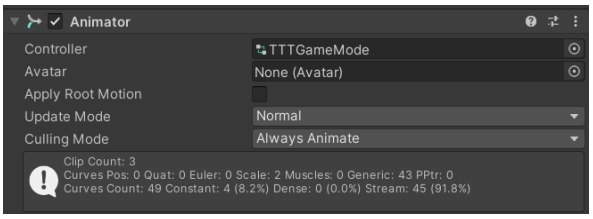
Local User is a special type of Manager which handles saving the latest state of the user to disk. The data structures are highly customizable and it uses serialization to save data to persistentDataPath.



Components

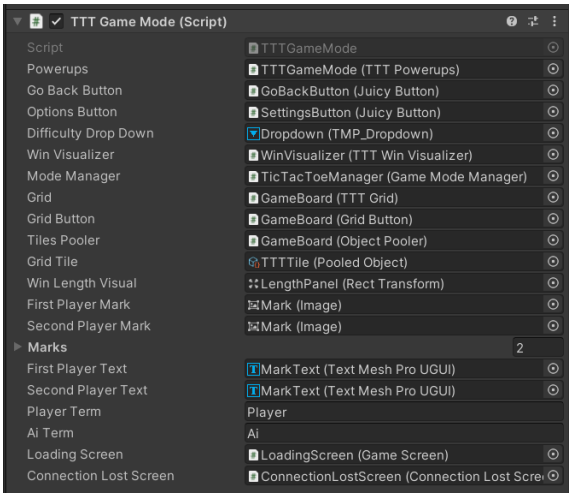
Game Screen





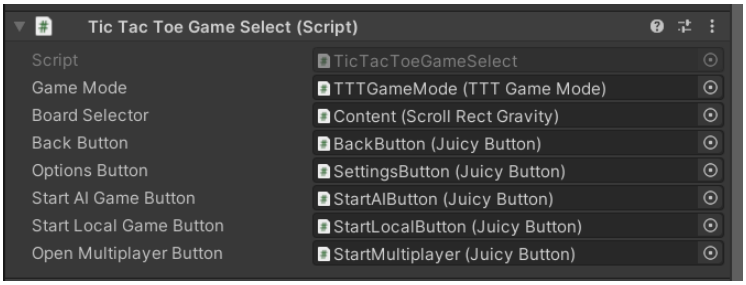
Each screen in the game contains the Game Screen component. It allows you to select this screen to open it. It can contain other Game Screens as children which will be enabled alongside it. Each screen also has an animator which tells the screen how it shows up and hides.

Tic Tac Toe Game Mode



Huge class which controls the flow of the Tic Tac Toe game. It handles the turns between AI and the player. It sets up the board based on the size of the selected board and initializes appropriate AI for the selected difficulty. It also handles game ending and triggers the end game screen to open. When in online mode it keeps track of the online status and stops the game if the player loses connection to the internet.

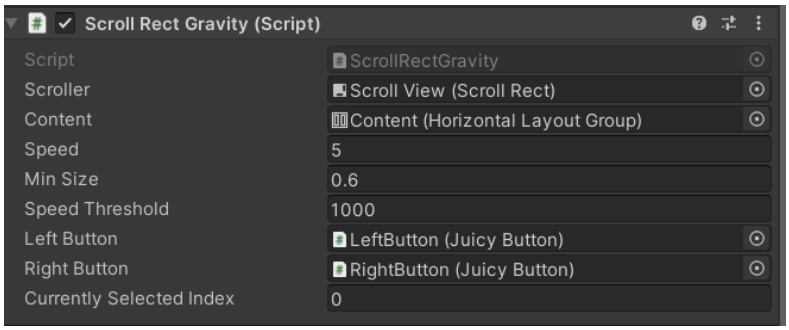
Tic Tac Toe Game Select



This component handles selecting game mode. The size of the board is selected based on a scroll rect which contains several board sizes in it. When starting a game the currently selected game board size is passed into the Tic Tac Toe Game Mode.

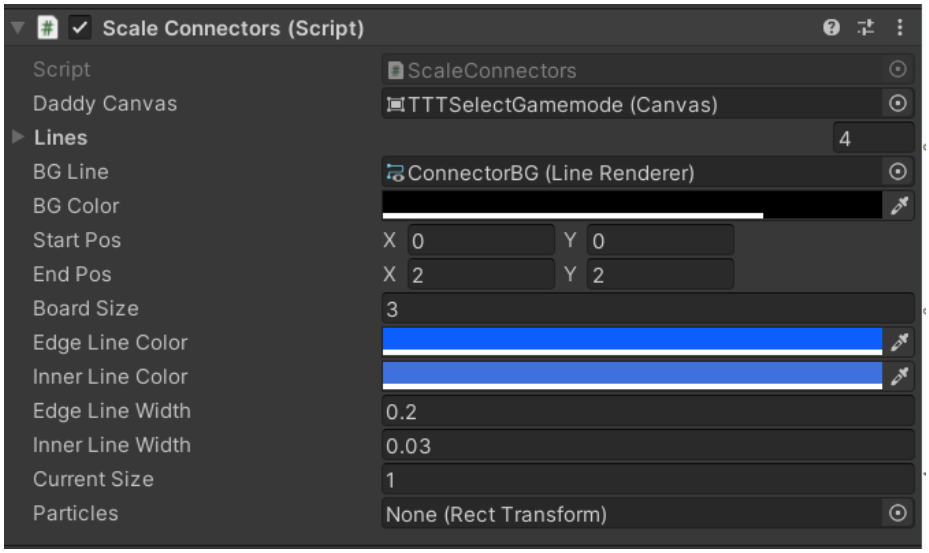
Scroll Rect Gravity

This custom component creates a scroll rect which is responsive both to user touch input and can be changed from the code. It allows for smooth change between game board sizes. It has a couple of settings regarding the speed of the scroll rect and naturally gravitates to one of the containing elements when moved by a finger. If flicked it will jump to the next option. In a way it works similarly to the scrolling mechanism of YouTube shorts or TikTok feed, but it is horizontal.



Scale Connectors

This component renders a line connection visualization. It uses several line renderers overlaid on top of each other to achieve this effect. It is fully customizable and can have particles



Game 2048

This component handles the whole game loop of the 2048 minigame. It is important that all spawned cubes are tracked by this Manager. The board does an animation transition when closing the minigame, so all tracked cubes' rigidbodies have to be frozen otherwise they will fly off.

Move Speed - How fast the current shooting cube moves when the player swipes along the screen with finger.

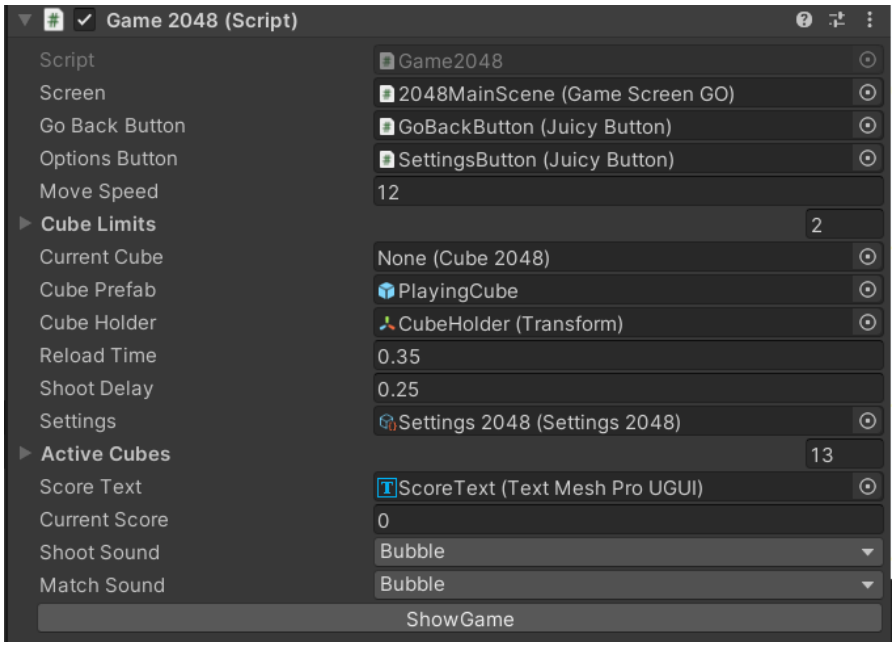
Cube Limits - The two positions in space between which the current shooting cube can move.

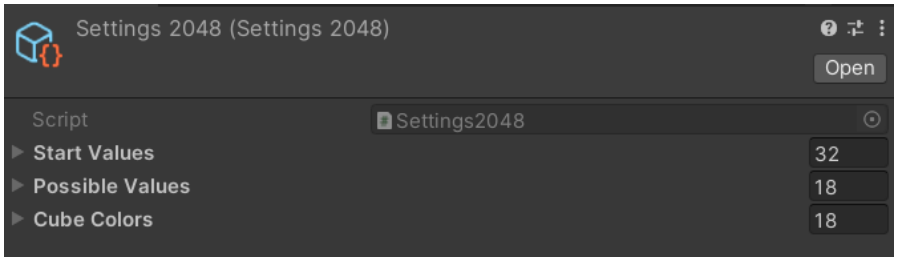
Reload Time - Time between shots.

Shoot Delay - Time between spawning a new cube and the time the player can shoot it.

Settings - Settings for the cubes are stored in a scriptable object.

Active Cubes - Cubes which are currently instantiated and tracked.





Cube 2048

A single cube which is used in the 2048 mini game. It contains a lot of settings for physics, animations, and visuals.

Gravity - The Y component of gravity which will affect this cube.

Appear Time - The animation of appearing takes this long in seconds.

Shoot Force - The force which is applied to the cube when the player releases the shooting cube.

Launch Force - When the cube is combined with another of the same value it will jump up with this force.

Sibling Force - When a cube is launched up by combining with another cube it will try to move towards other cubes with the same value to create a chain reaction.

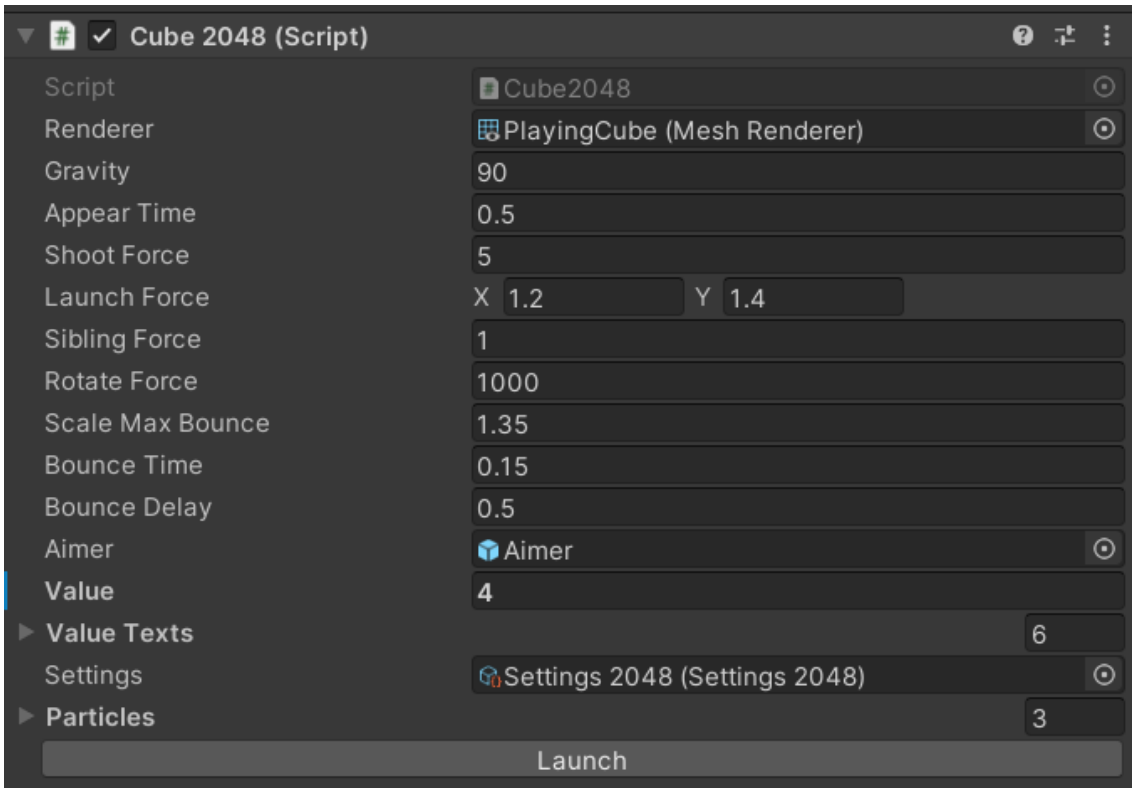
Rotate Force - When a cube is launched into air it will try to randomly rotate with this force.

Scale Max Bounce - When a cube is launched up it will do an animation of increasing its scale. This is the max value it will scale up to.

Bounce Time - How long will the animation of scaling up and down take time.

Bounce Delay - How much is the animation of scaling up and down delayed compared to when the cube starts launching into air.

Value - The value of this Cube.



Random Bounce

This component makes anything bounce by modifying the scale of the game object. It does the bounce along the provided curve. The bounce has defined duration and maximum offset from the base scale of the object. The bounce can only happen after initial delay and then repeats at a random time which lies in the interval bounce random time span.