RPG INDICATOR

Overview

RPG Indicator is the perfect tool to add optimized indicators to your game. The tool is designed to be simple, save you a lot of development time and intuitive to use. The tool is battle tested and currently in use for my game Brinefall and comes with an RPG Builder integration.
Perfect for your RPG, MMO, Roguelike or Strategy game.
RPG Indicator is a shader using the Unity projector and is compatible with Unity HDRP 2020.3+ and URP 2021.3+.

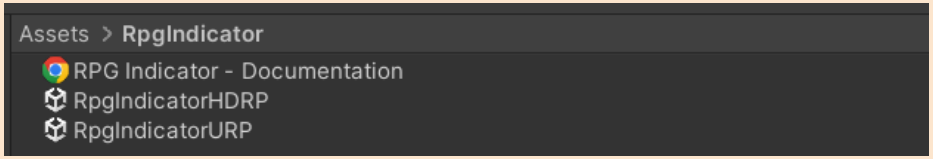
Features

- **Support uneven terrains and slopes**
- **Custom texture and style**
- **Simplified workflow, setup everything only once**
- **Made with Shader Graph**
- **Can be used in one line of code (sample provided)**
- **RPGB Integration**
- **Rigorously tested in a real game**

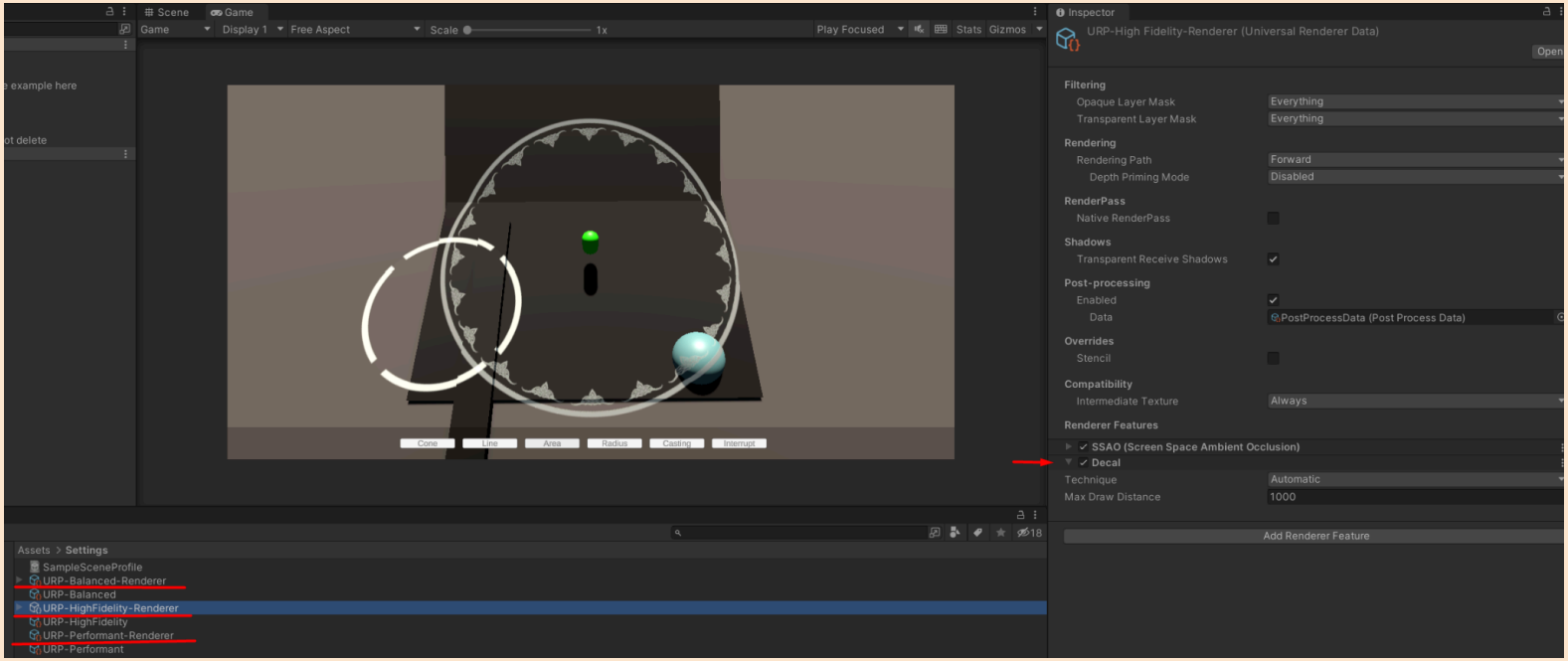
Get Started

RPG Indicator is made in 3 parts. The indicatorData (scriptable object), the StatusIndicator (GameObject) and the materials.
There is a demo scene in RpgIndicator/Demo you can try. In this scene you will have some indicator examples and code samples you can use.

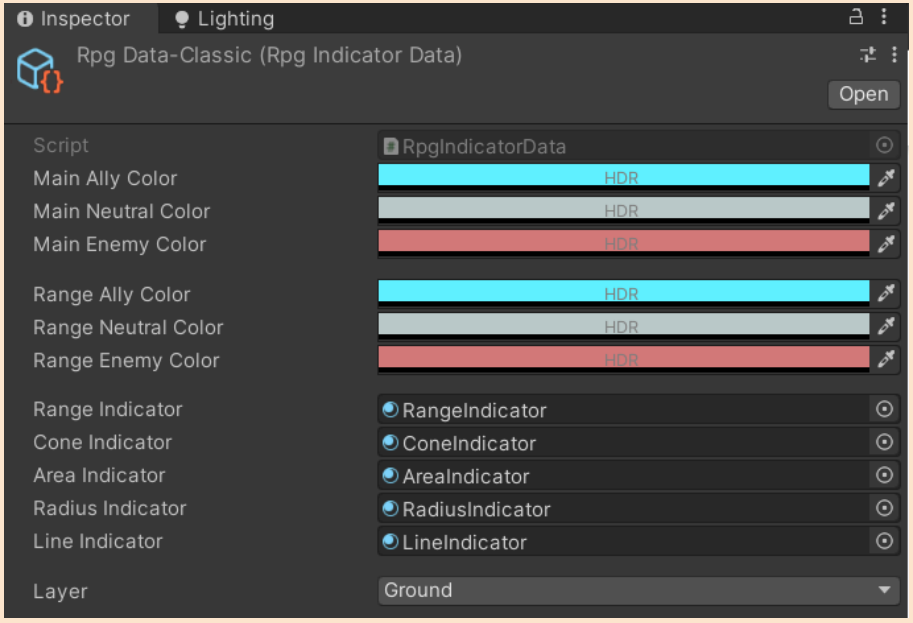
First, install the RPG Indicator relevant to your pipeline (URP or HDRP)



In **URP**, add *Decal* to your renderer for each quality setting you have.



The IndicatorData is a simple scriptable object where you can set all your indicators at once.
The first part is the color settings where you can choose the colors to use of the main part and range part of your indicator. This will tell the tool to use a specific color depending on the alignment of the unit using it.

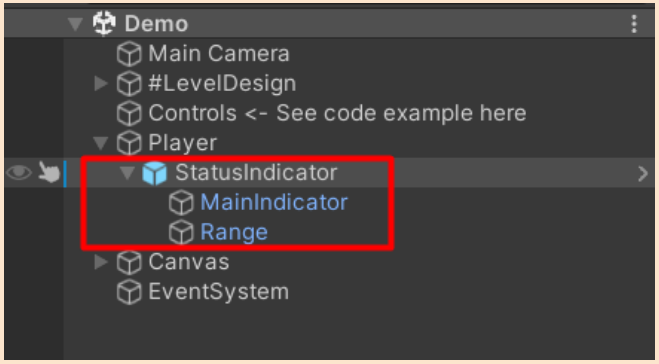


The second part are the materials to use, in it you will be able to tweak the style a bit further and change the textures.

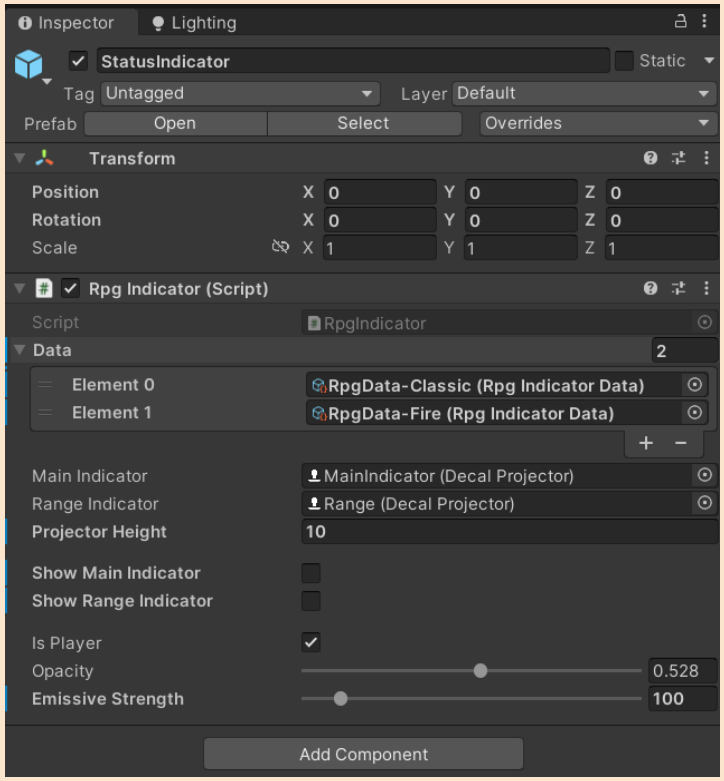
Last part tells the tool which layer to use for the ground detection mostly used for the mouse detection.

Note that you can create multiple IndicatorData and choose which one to use.

The StatusIndicator is a gameObject you place as a child of any player / enemy or object you want.



The first part of this RPG Indicator is the **Data** section. This is where you will insert The IndicatorData previously created and it will tell your indicator what kind of style To use.

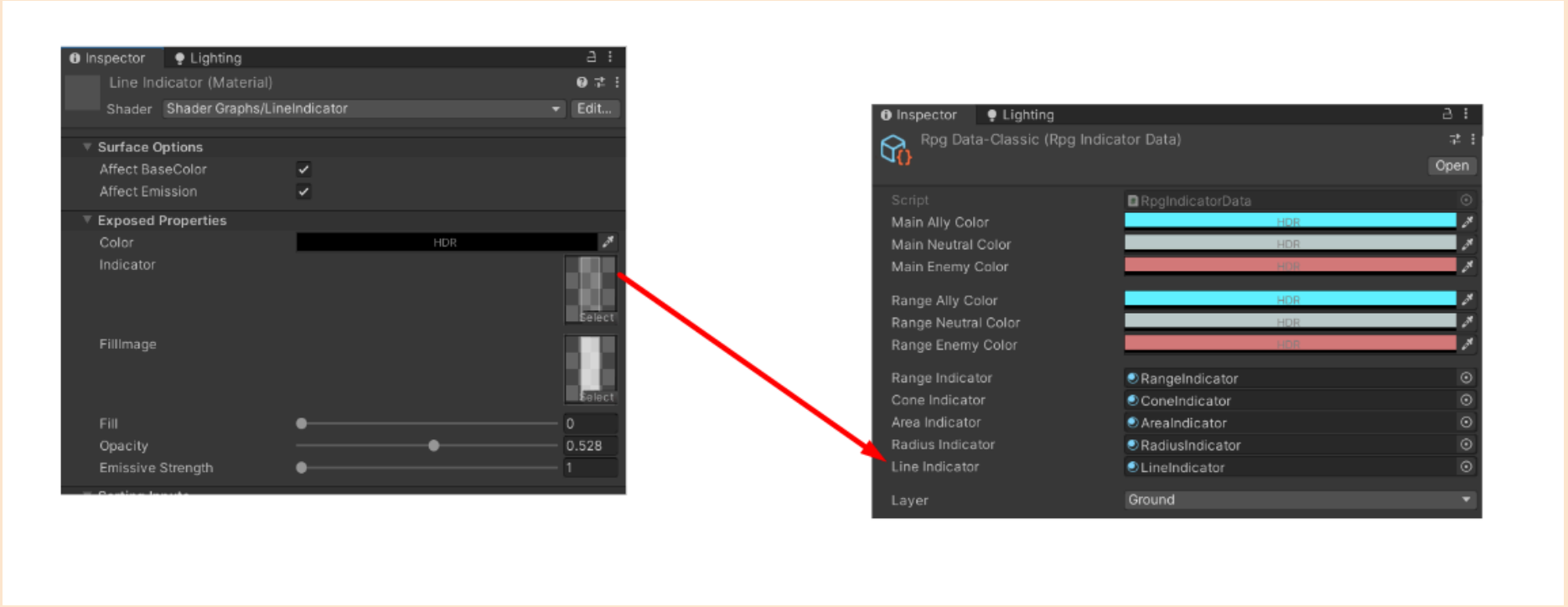


Projector Height: Determine the height of the projector.

isPlayer is to check if this StatusIndicator is the player. It is used in the code to determine certain behavior of the tool.

The last part is for customization
Opacity: Increase or reduce the opacity of the indicator.
Emissive Strength: Allow you to add emissive to the indicator.

The material only has one important section and it is for the textures. There is one material to prepare per indicator type (Range, Cone, Area, Radius and Line)
When a material is set and ready you can add it to your RpgIndicatorData so that the system know which one to use.



Note that you don't have to touch the Fill / Opacity / Emissive Strength parameters because they are controlled by the tool.

RPG Builder Integration

Rpg Indicator is fully integrated with RPGB and you can do everything you need without writing a single line of code.

First step: Install the integration package in Rpg Indicator/RPGB Integration.

You can also download it here

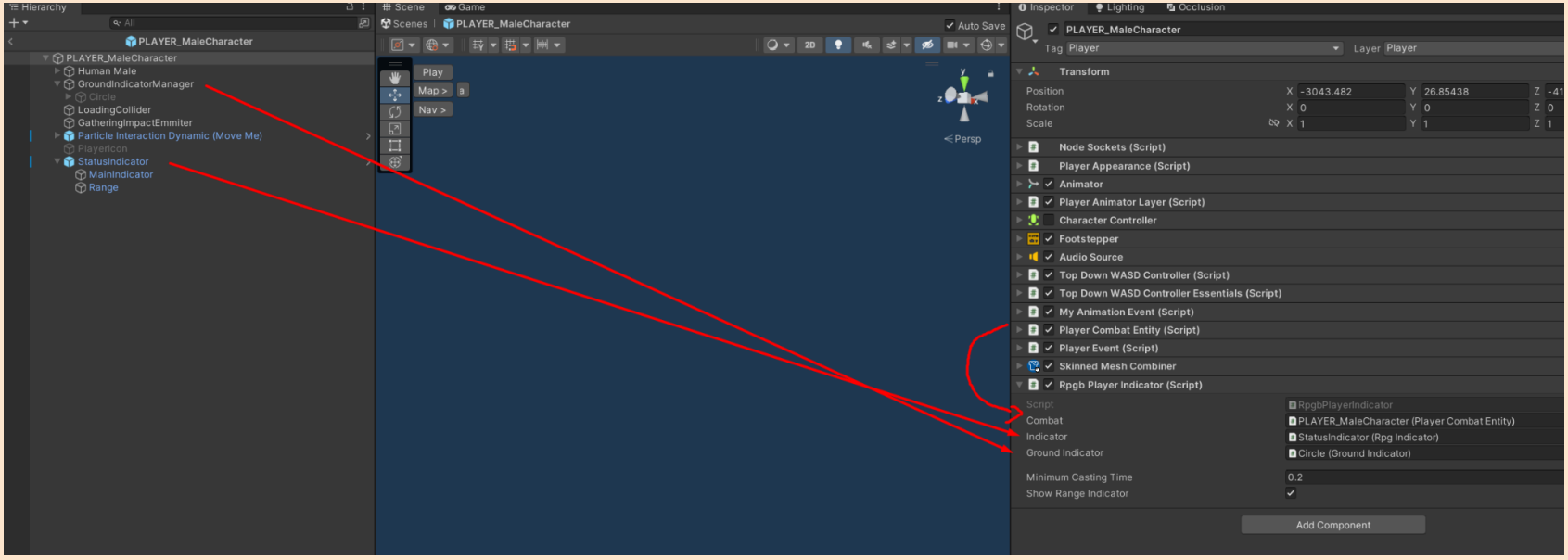
Installing the player:

First, add the script RpgbPlayerIndicator to all your playable races.

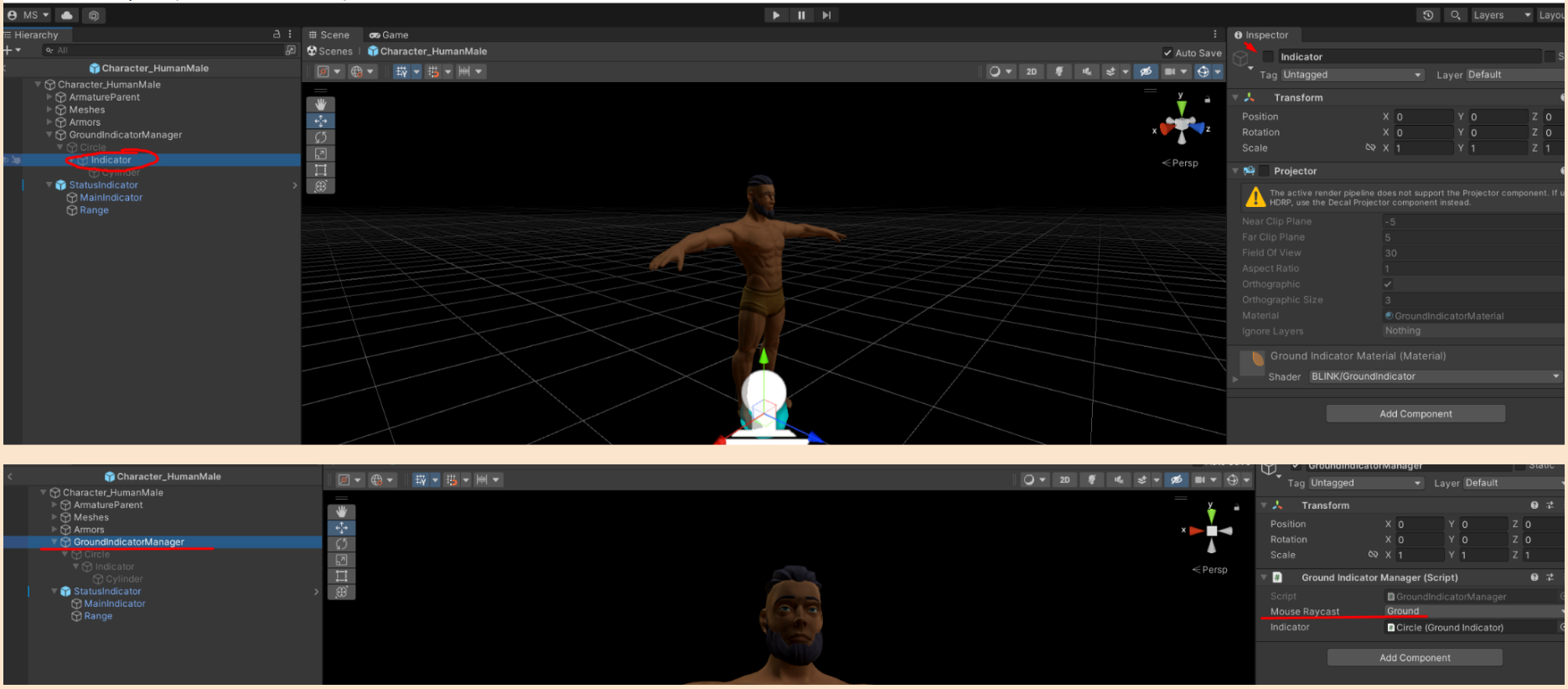
Once this is done, the integration will require the Player Combat entity to be added in “Combat”.

The StatusIndicator object needs to be added as a child of each race and added inside the script in the indicator section.

And finally the original RPGB ground indicator needs to be added in.



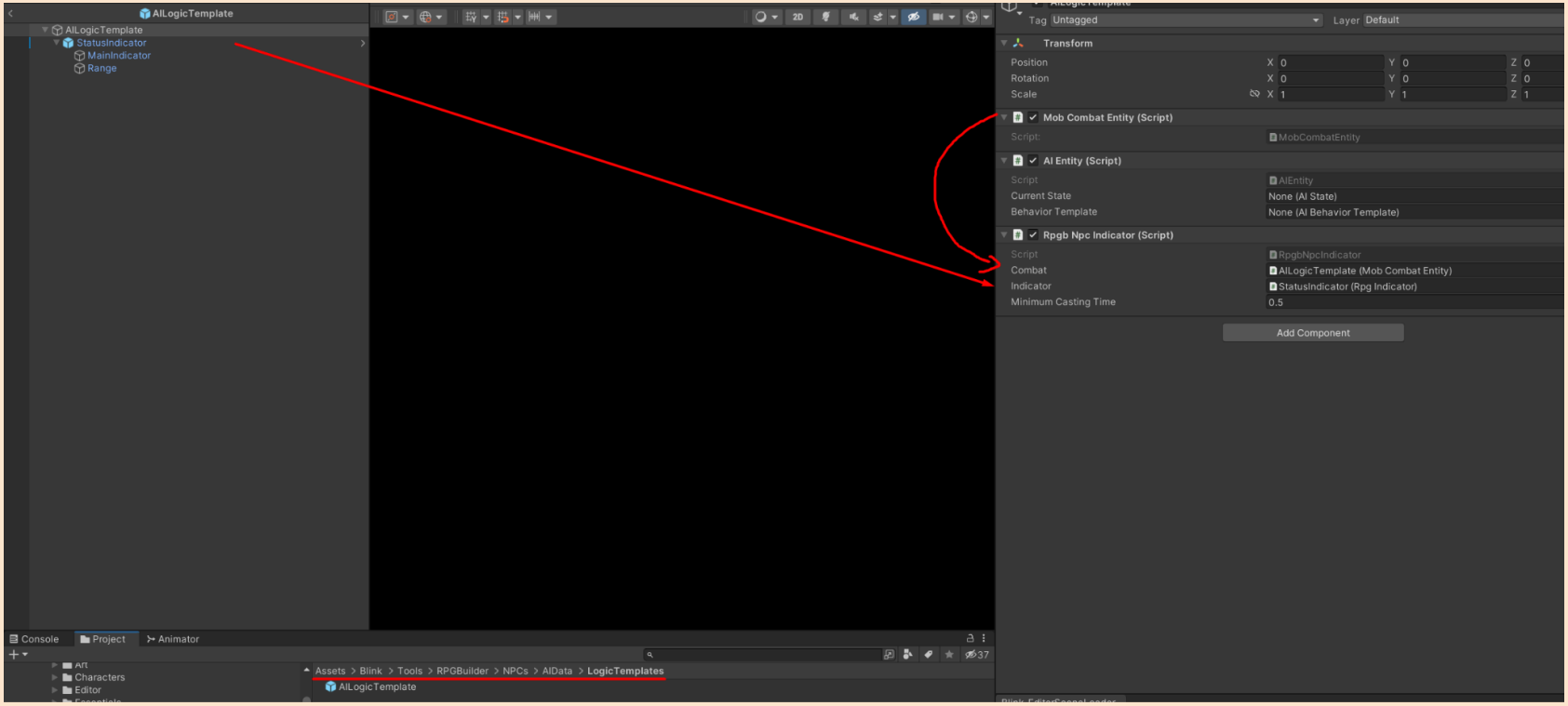
Be sure to disable the basic RPGB Indicator, and set the Mouse Raycast to the layer you want to use.



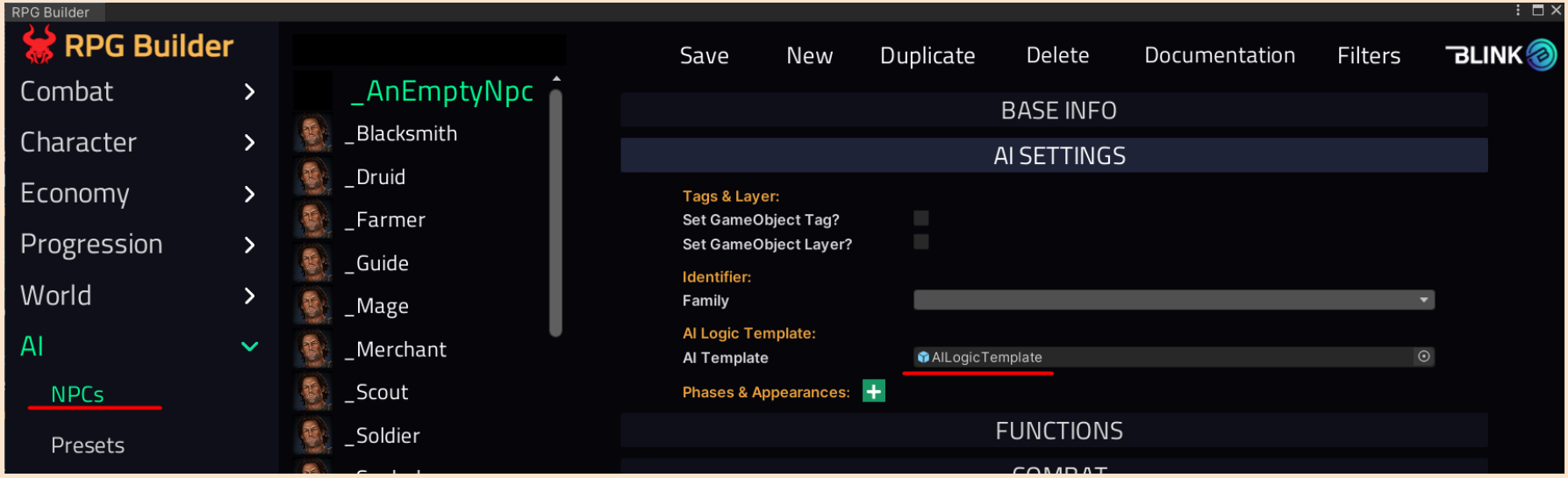
Minimum Casting Time is an additional option that allows you to ignore some abilities that are just too short to be worth being shown. In that case, every ability with a casting time shorter than 0.2 sec will be ignored and not show any indicator.

Installing the NPCs: Go inside *Assets/Blink/Tools/RPGBuilder/NPCs/AIData/LogicTemplates* , in there you will have a small GameObject AI Logic Template. You can either use it or copy it.

Add the StatusIndicator to the AI Logic Template and add the script RpgbNpcIndicator to the object and then add the required Mob Combat Entity, indicator and the casting time just like we did with the player.

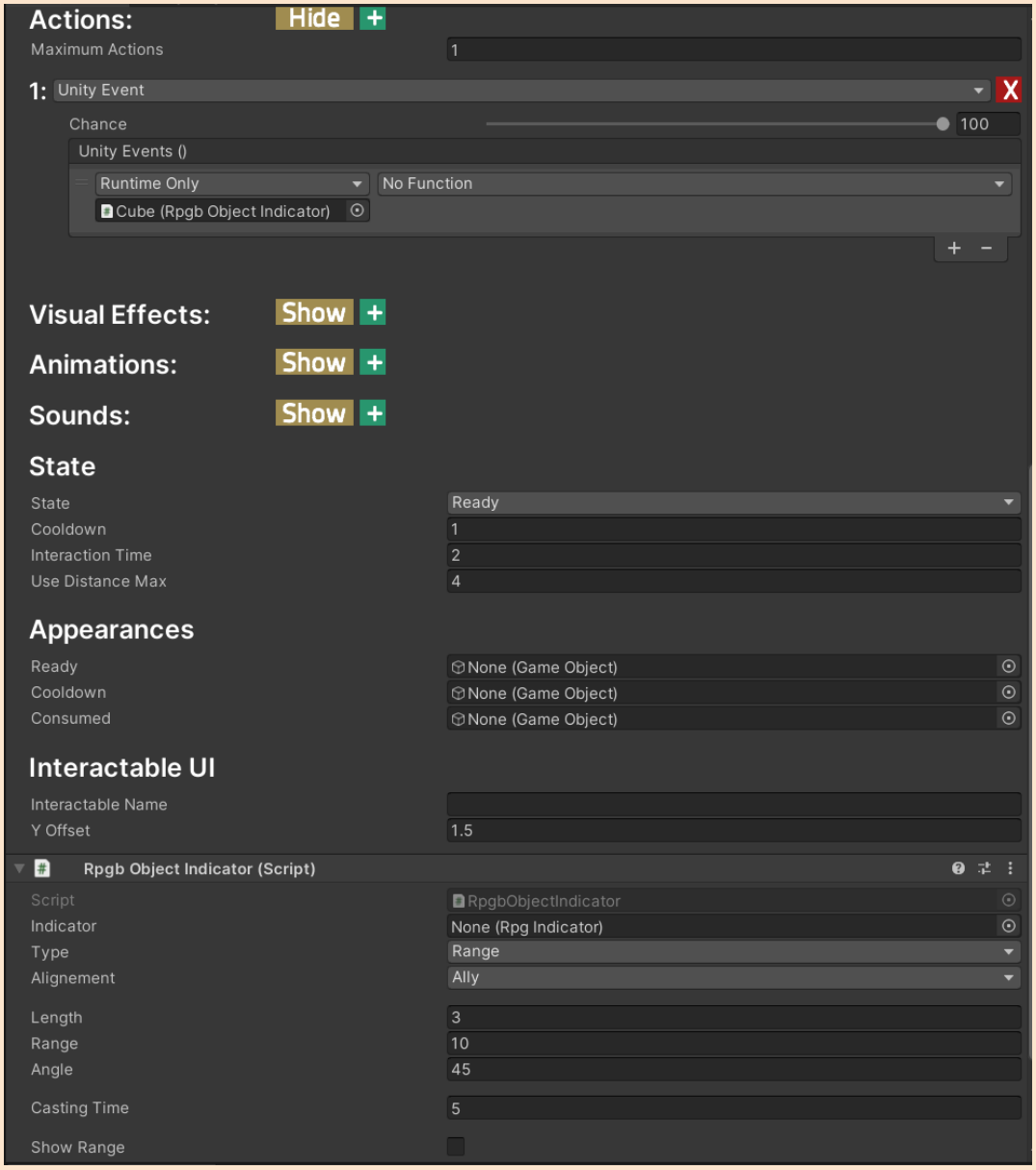


Then go inside the RPGB editor and go to Ai / NPCs. Inside you will see an AI Logic Template section. Simply make sure that your AI Logic Template is in there and save.



Finally, the indicator can also be activated via an **interactable object.** Add the RpgbObjectIndicator script to an intractable object and add an “Actions” of type Unity Events().
Simply drag the script into that event and you can activate it via the function ShowIndicator() or Cast()

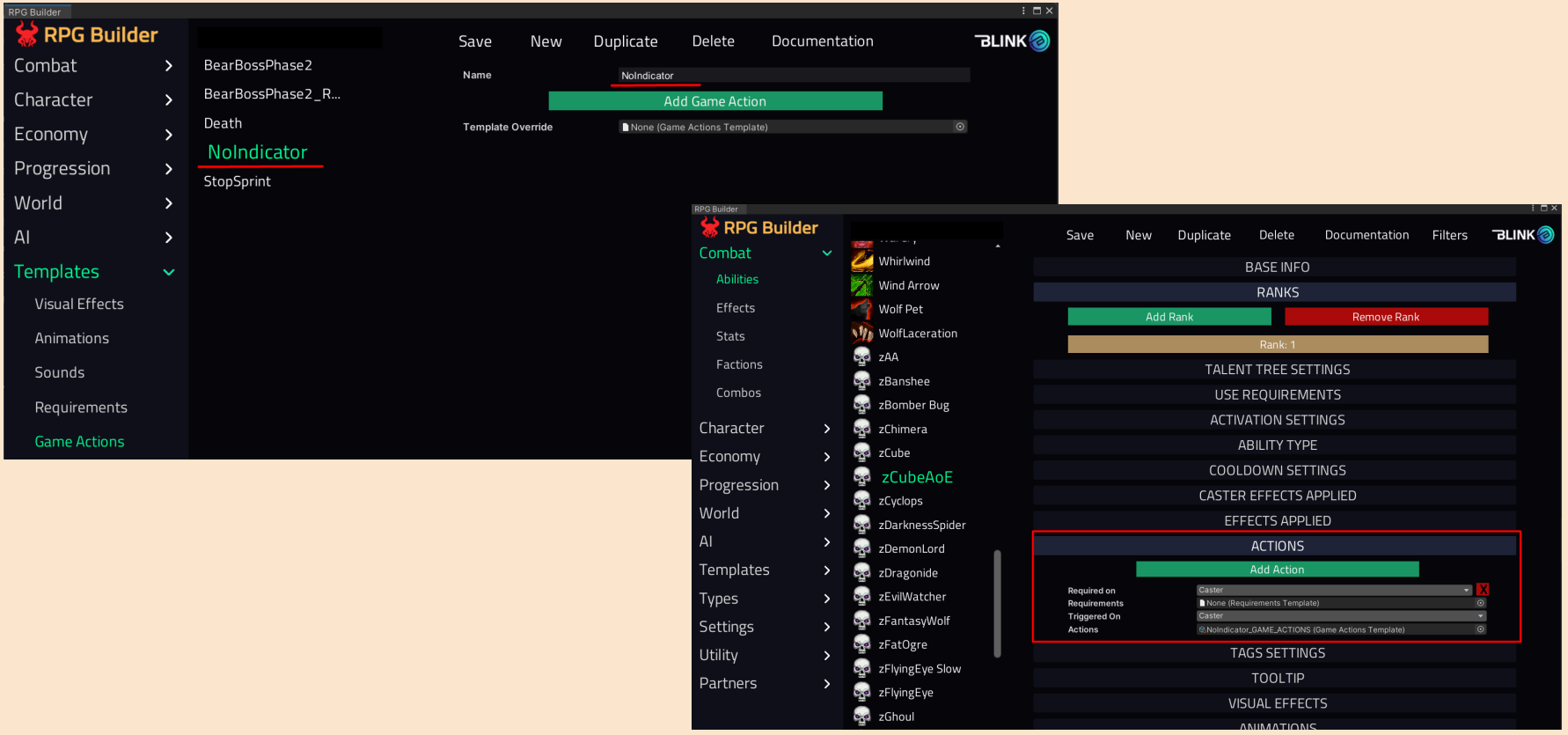
Note that the script doesn’t have to be on the same object and could activate one or multiple other indicators.



Ignoring indicator

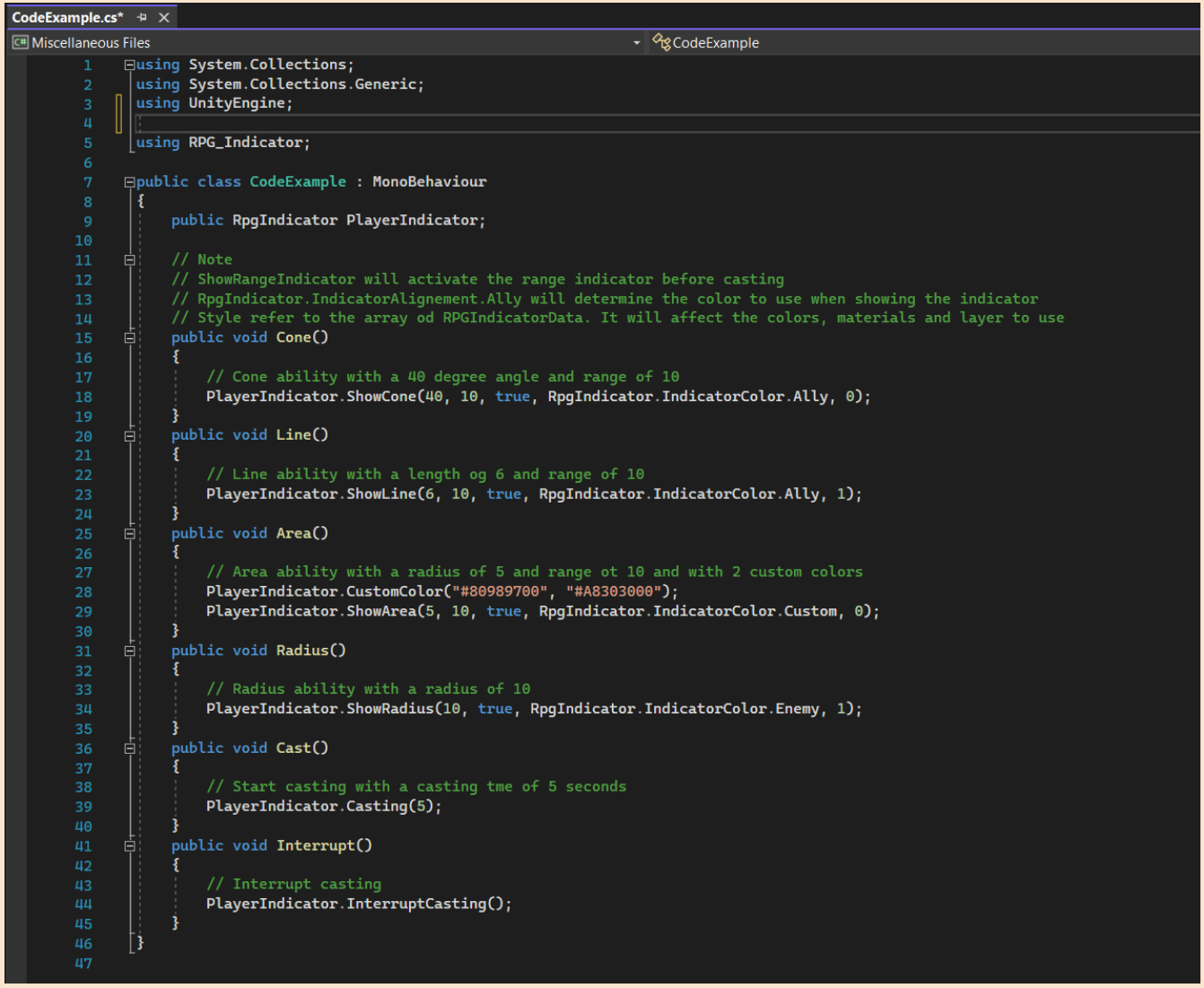
The integration allows you to prevent RPG Indicator from being used with a specific ability. This could be useful for preventing some abilities like auto-attack to always trigger an indicator.

In order to do this you need to create a Game Actions called ***NoIndicator*** and add it to your ability to be ignored.

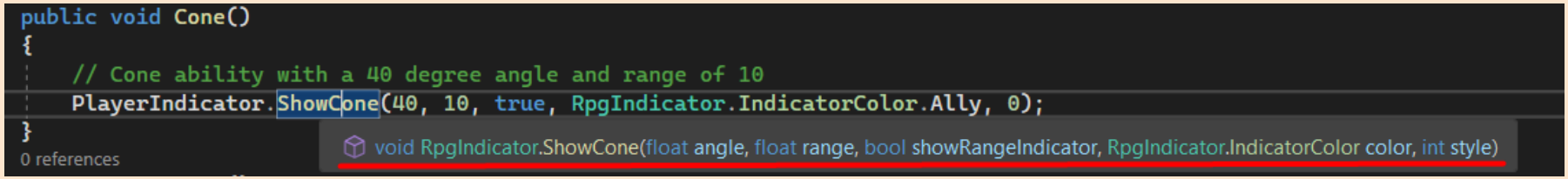


Coding with Rpg Indicator

The code is made to be very easy to use and some templates are even provided in the CodeExample script used for the demo.



For more detail on each function and how to fill them, you can simply mouse over a function and it will tell you what it need



In the example above, the function ShowCone will require the angle of the cone, the range, if you need to show the range indicator, the alignment towards the player and the style to use. The style refers to the IndicatorData inside the indicator GameObject.

The available functions are

ShowCone()

ShowLine()

ShowArea()

ShowRadius()

Cast()

Interrupt()

FAQ

My indicators draw above my character

HDRP: You can either disable “receive decal” on the material or use decal layers in your HDRP settings to use layers and have full control on what will receive decals or not.

URP: Unfortunately URP doesn't support layers like HDRP does. It is a limitation of URP. What I could suggest is to reduce the height of the decal to make it less visible.

Support

If you have any question, don’t hesitate to contact me at michaelsimard@qwertystud.io