

Thanks for purchasing Gears and Levers package!

If you have any suggestion or need support, feel free to contact me on: zefaistos@live.com

ANIMATIONS

Some example animations are included with every prefab. If you want to use the example animations, make sure to activate the Animator component of the desired prefab, and disable the rotation script attached to the gear wheel prefabs, in order to avoid any movement conflict. More information about the rotation script below.

To play an animation by script, you can navigate to the models>animators folder and double click the corresponding Animator to open the Animator tab. Now on the Animator tab, you can see the name (the grey rectangle) of the animation to be called.

As an example, if you want to cast an animation for any of the gear wheel prefabs, the code would be the following:

```
gameObject.GetComponent<Animator>().Play("Rotate");
```

The gameObject in this case is the root object with the Animator applied. The gear wheel prefabs use a single animation. If you want to cast the same animation multiple times during the same runtime, you first may need to call the idle state (same code as above, just replace "Rotate" for "Idle"), and then call the above code for the animation. Otherwise, the animation would not play again, because it's still in the animated state in the last frame. Another solution would be creating a transition between both states, but in this case not every gear wheel will return to the same start rotation, and the player would visually see the transition between the end of the animation and the idle state, which is not desirable.

For the Levers and Mechanisms, there are multiple animations included, so you can call the animations in sequence without the need to call the idle state first. The animations are named as follow:

- Levers: Activate/Return
- Mechanisms: Rotate1/Rotate2/Rotate3/Rotate4

Same code as above, just replace the animation name in the function.

Of course, those are only simple example animations. You may want to create your own animations or scripts to best fit the player interactions in your game or project.

ROTATION SCRIPT

A simple rotation script is included for the gear wheel prefabs. But instead of the root object, the script is attached to the child objects. Each individual gear wheel has the script attached in order to have an independent rotation.

If you want to disable the script, select each child gear wheel and disable or remove it. If you want to change it's constant rotation, you may need to calculate the speed for each gear wheel in order to match their rotation.

GEAR ROTATION CALCULATION

When you have gear wheels with different numbers of teeth/cogs, the gear with the smaller number of teeth needs to rotate faster. To calculate how faster it needs to be, you just have to count the number of teeth on both gears, and make a simple division equation. The result will be the additional rotation that the smaller gear needs compared to the larger one.

For example, in this package the Gear1 prefab includes 2 gear wheels: one with 14 teeth and another one with 8 teeth. In this case $14 / 8 = 1.75$. Now we now that the smaller wheel must rotate 1.75 times the rotation of the larger wheel. As you can see in the script values, while the larger wheel is rotating at a speed of 40, the smaller gear is rotating at a speed of 70 ($40 * 1.75 = 70$). The value is also negative to rotate at the opposite direction.

This same concept applies for animations. If you rotate the bigger gear wheel 360 degrees over a period of time, the smaller gear must rotate 630 degrees over the same period of time ($360 * 1.75 = 630$).

You can mix any gear of any size or number of teeth (if it makes sense visually). It's important to remember to count the number of teeth and make a division to discover the additional rotation required in order to match the movement of the multiple gear wheels.