

FISHING GAME TOOL

Fishing System Development Tool for Unity

USER GUIDE

Release 1.2

10.2023

Created by Paweł Mularczyk

This manual, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this manual is furnished for informational use only, is subject to change without notice and should not be construed as a commitment by its authors. The author assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual.

FISHING GAME TOOL

Table of Contents

Getting Started

Overview.....	3
---------------	---

Fishing Rod

Fishing Rod Model.....	4
Configure Fishing Rod.....	7

Fishing System

Configure Fishing System.....	14
Configure The Fishing Spot.....	20
Testing.....	23

Additional components

Character Movement.....	25
TPP Camera.....	26
Hand IK.....	27
Simple UI Manager.....	28

FISHING GAME TOOL

Overview

The **Fishing Game Tool** stands as a versatile asset, facilitating the effortless integration of a fishing system into a wide range of project types. Whether your endeavor involves crafting a survival game, a casual production, or an intricate RPG, this tool furnishes the essential framework. It matters not if your game takes on a top-down perspective or immerses players in a first-person viewpoint (FPP); this tool furnishes the foundation for crafting captivating fishing encounters. Tailor fishing mechanics, loot, and the odds of acquisition to harmonize these interactions seamlessly with your game's essence.

Through its adaptability and user-friendly design, the **Fishing Game Tool** empowers the realization of a gratifying fishing experience that harmoniously aligns with the tone and aesthetic of your game.

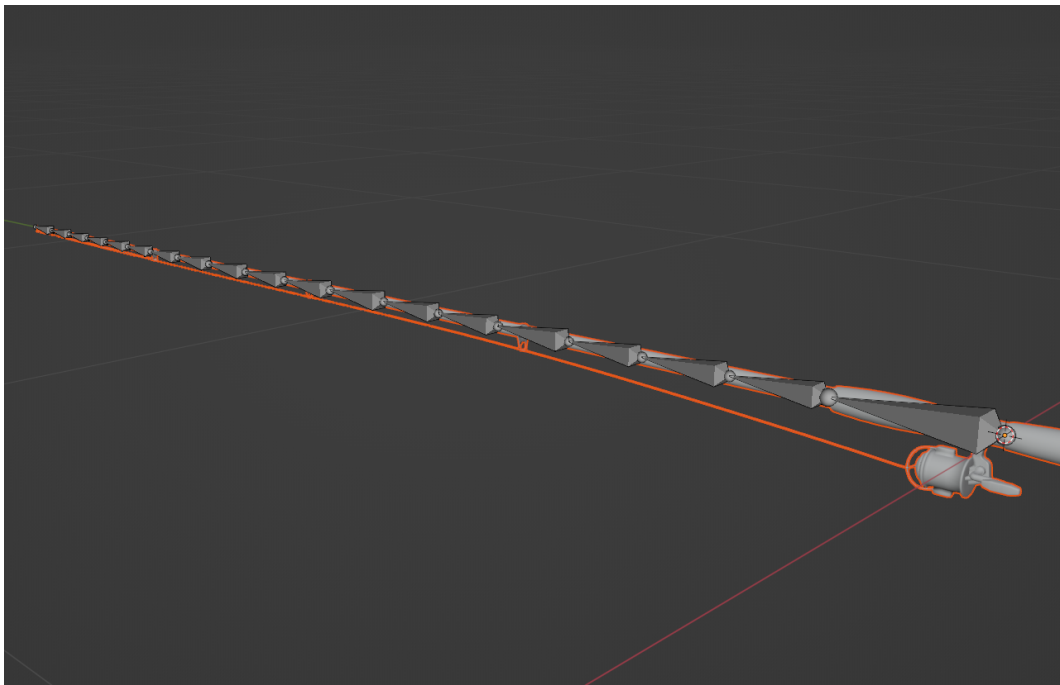
FISHING GAME TOOL

Fishing Rod Model

We initiate the preparation process for a fishing rod model by constructing a skeletal framework for the rod. This framework will serve as the basis to define the extreme bending positions of the rod. As we proceed, these positions will be harnessed to create a controlled bending system for the rod in later stages.

1. Add a skeleton to the rod model.

To ensure accurate bending, the framework should encompass a minimum of 15 bones. The bones situated nearer to the grip should possess greater length and gradually taper towards the rod's end.

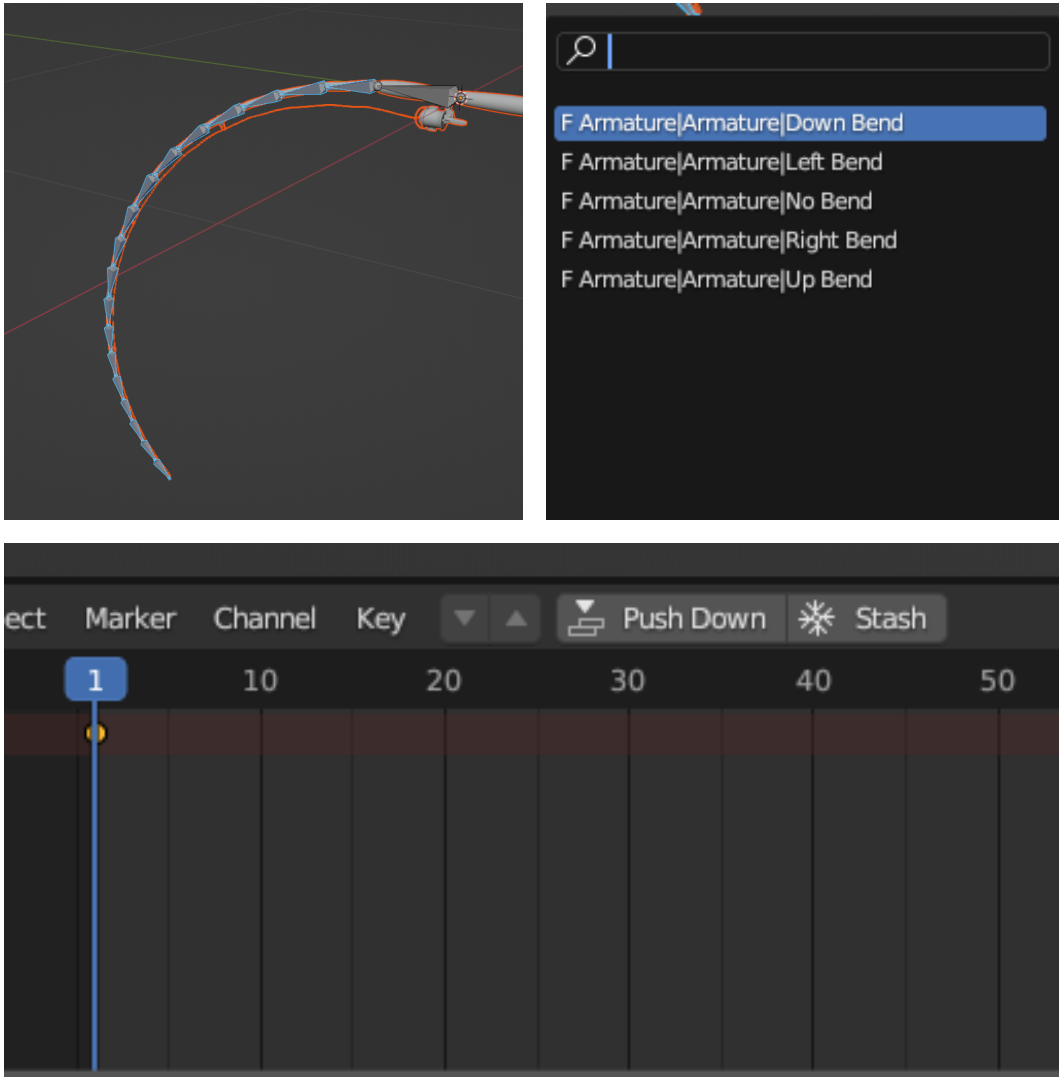


2. Creating extreme bending points.

At this stage, we will be establishing the extreme bending points of the rod. This will involve four bending points on each side, along with one point representing zero bending.

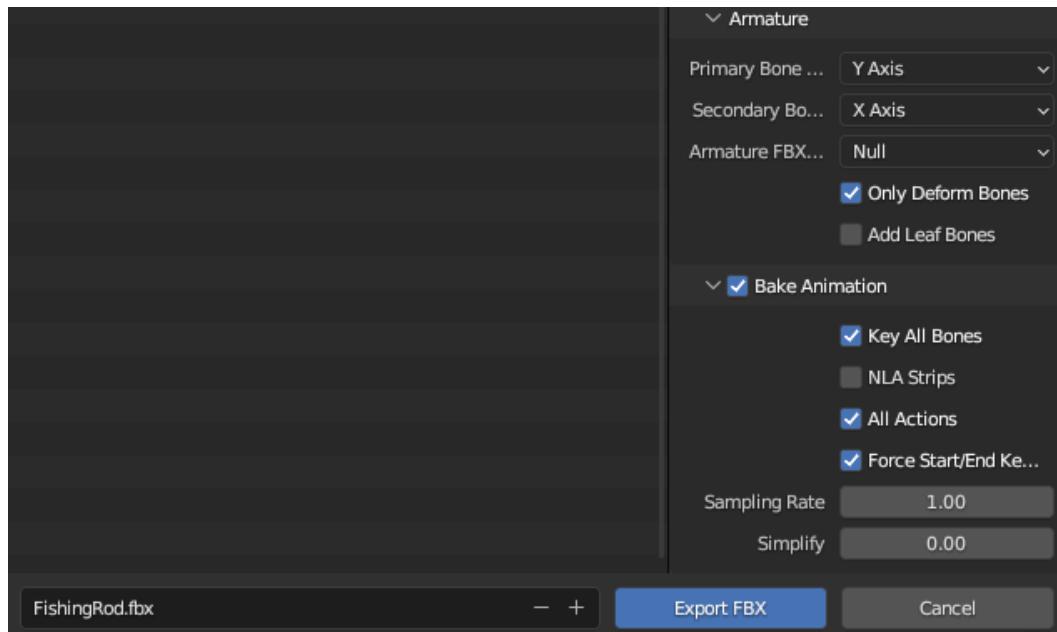
The bend should have only one animation frame.

(Transform Pivot Point set to Individual Origins in Bledner)

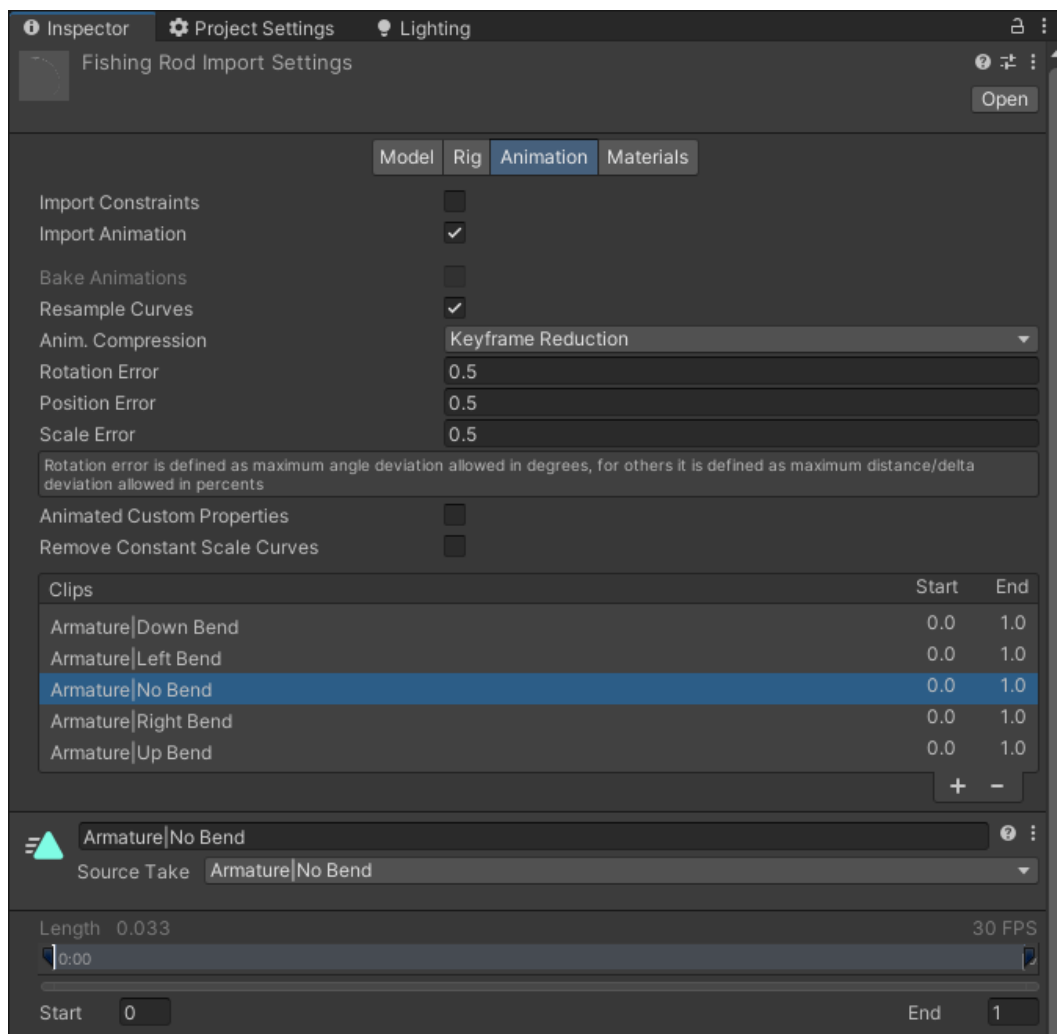


3. Export model and import to Unity.

Export as an fbx file.



Import the model into Unity. Within the Animation tab, generate five animation clips and allocate the bending animations to these clips.



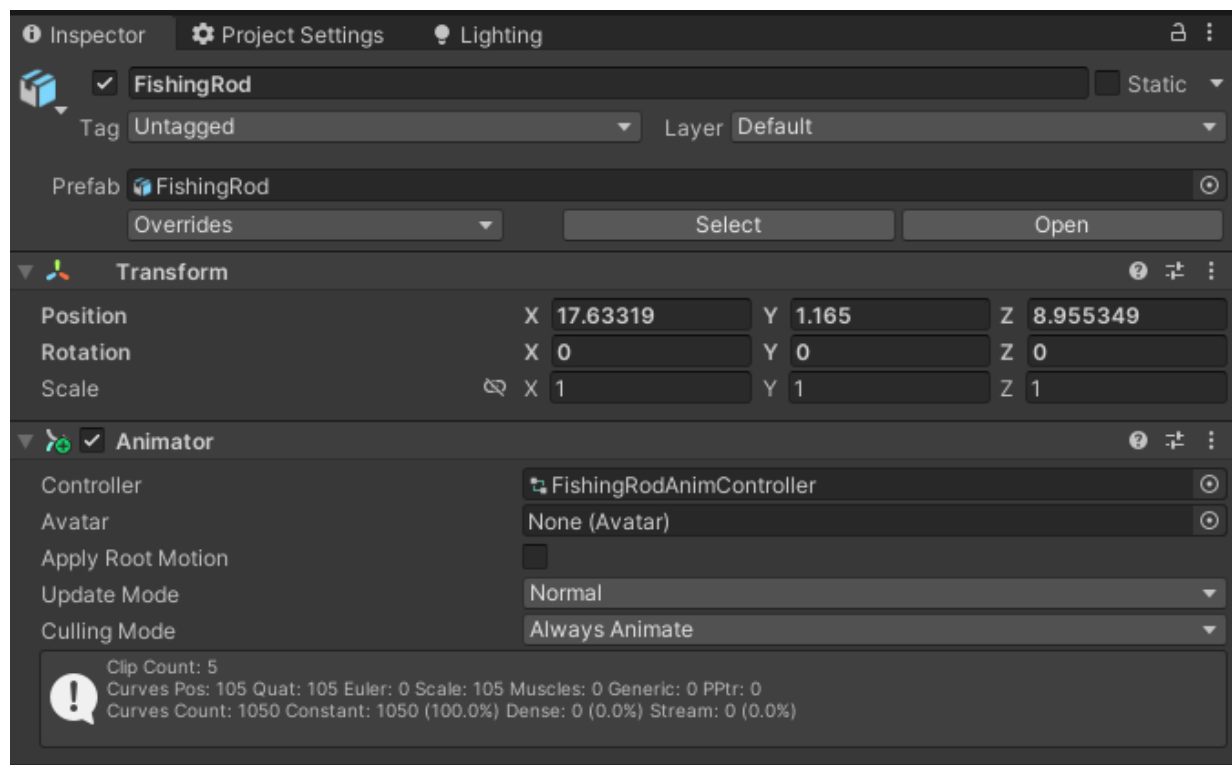
FISHING GAME TOOL

Configure Fishing Rod

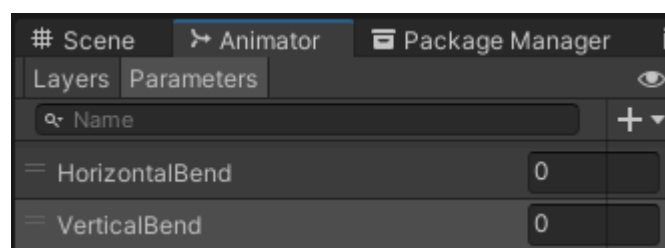
During this phase, our attention will be directed towards configuring the fishing rod.

1. Configuration of the fishing rod animator.

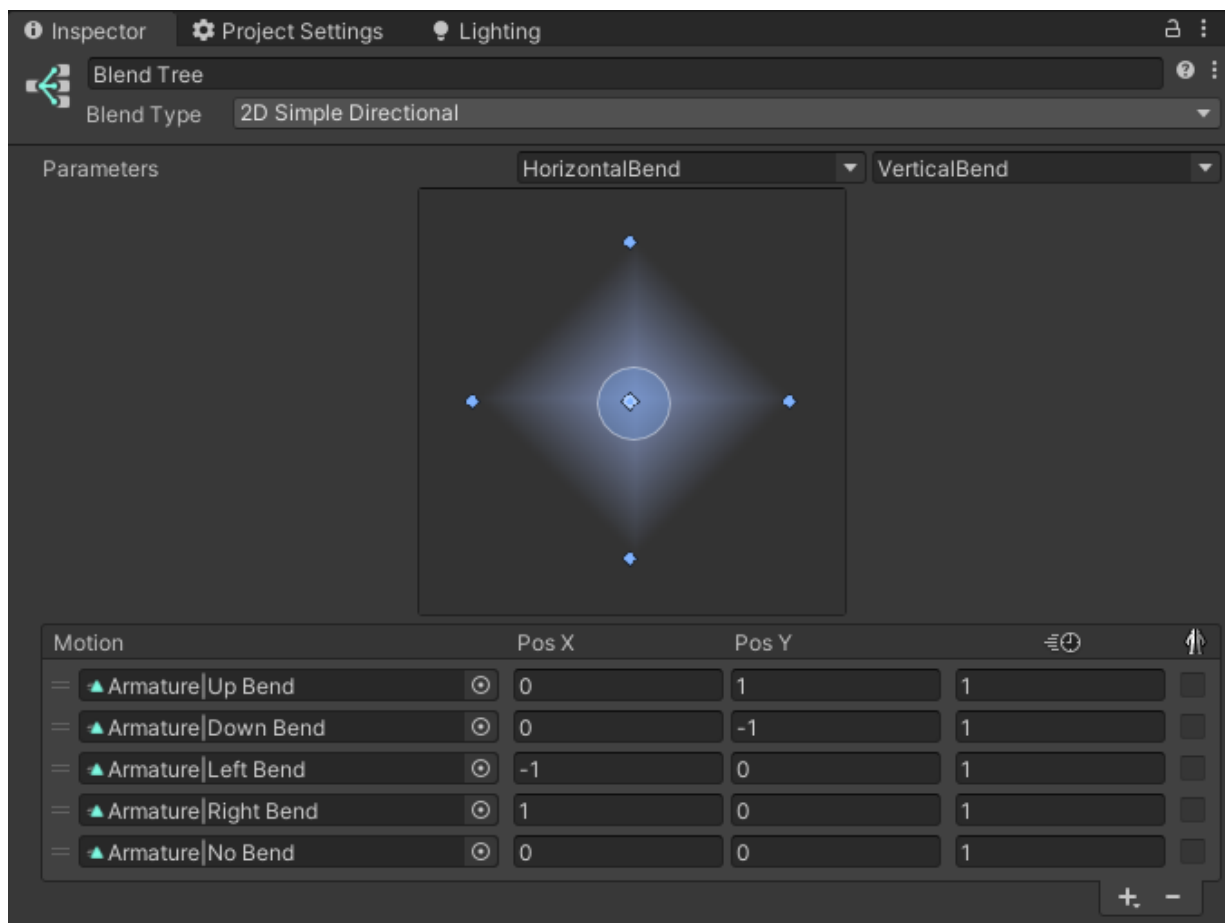
Initially, we will incorporate an animator component into our fishing rod, in conjunction with the animation controller.



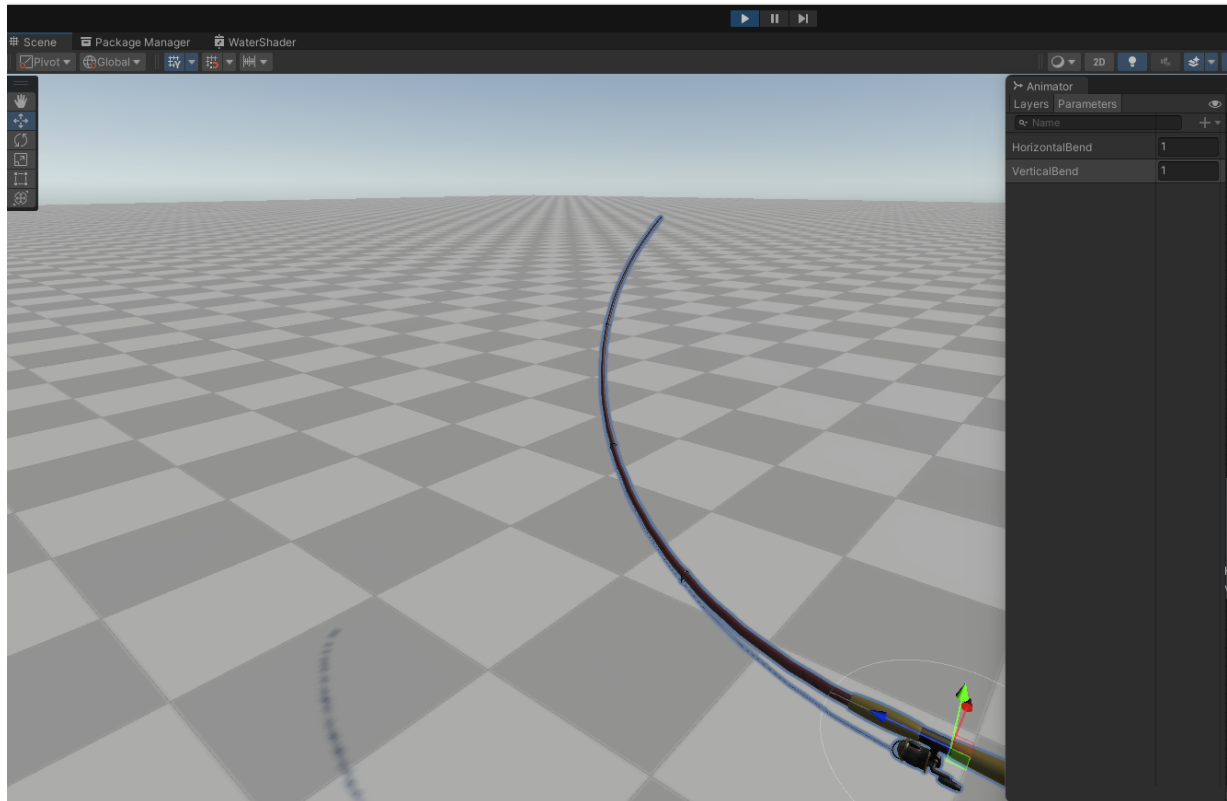
In the Animator panel, specifically within the Parameters tab, we introduce two parameters of the float type: **HorizontalBend** and **VerticalBend**.



We are in the process of generating a new blend tree and designating it as the default state. Within the blend tree configuration, select **2D Simple Directional** as the **Blend Type**. For the first parameter slot under **Parameters**, assign **HorizontalBend**, and for the second parameter slot, designate **VerticalBend**. Within the **Motion** section, incorporate five elements and position the animations we previously included for the fishing rod. Align the **Pos X** and **Pos Y** parameters to correspond with the directions of the rod's bending.



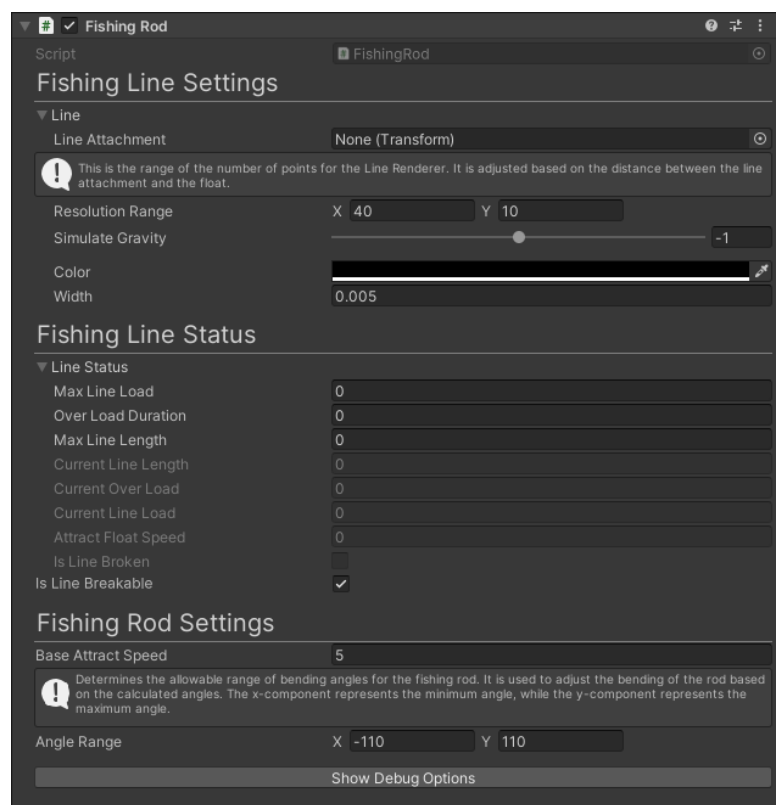
Upon successfully accomplishing these actions, the bending functionality of the fishing rod should be operational. You can manually verify this by altering the values of **HorizontalBend** and **VerticalBend** and observing the results.



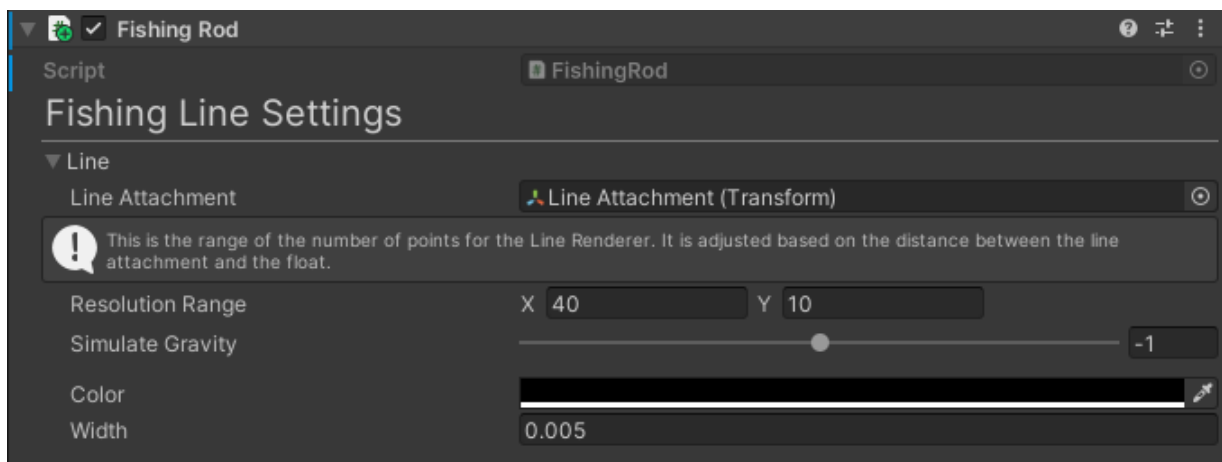
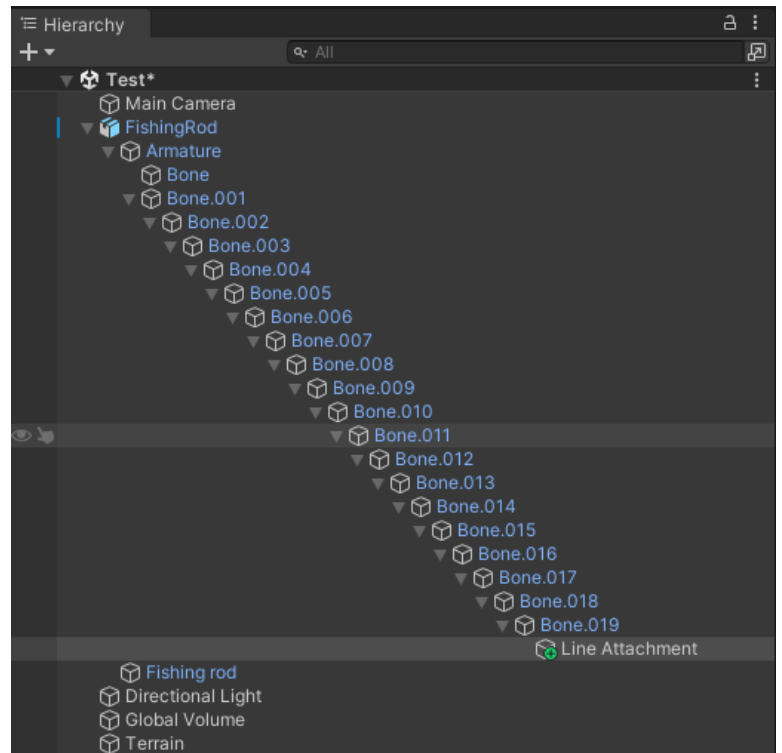
2. Configure Fishing Rod.

To add the Fishing Rod component, navigate to **Add Component -> Fishing Game Tool -> Fishing Rod**.

Simultaneously, the **Line Renderer** component will also be appended. This component is employed for visualizing the fishing line.



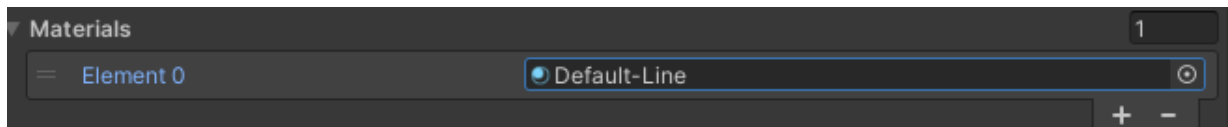
Let's initiate the configuration process from the **Fishing Line Settings** section. Under **Line**, you'll find elements dedicated to configuring the fishing line. Begin by designating an empty Game Object as the point of origin for the line. This Game Object should be positioned at the terminus of the fishing rod's final bone. Subsequently, employ the created Game Object as the **Line Attachment** within the **Fishing Rod** component.



Resolution Range defines the range of points in the line renderer based on the distance between the fishing float and the fishing rod. Value X determines the number of points close to the rod, and Value Y determines the number of points when the fishing float is distant. The point count changes smoothly based on the distance.

Simulate Gravity determines the force acting on the fishing line. The lower the value, the more the line sags. A value of 0 makes the line taut.

Color defines the color of the fishing line. To make it work, you need to add a material to the Line Renderer. **Width** determines the width of the fishing line.



Fishing Line Status stores information about the durability of the fishing line and its current state. **Max Line Load** determines the maximum load the line can withstand. This value should not be less than 10. It significantly affects the fishing difficulty. **Over Load Duration** specifies the time after which the line will break if the Max Line Load is exceeded. **Max Line Length** parameter defines the maximum length of the fishing line. When a fish is caught, it will not swim beyond this specified maximum line length. The fishing float on ground will move in the direction of the fishing rod to prevent exceeding this distance. When casting the float, setting this value too low will result in the float stopping prematurely. **Is Line Breakable** indicates whether the line can be broken at all. Deselecting this option will not break the line, but after exceeding the catch limit, the fishing action will still conclude. However, the line won't need to be re-rigged.

A screenshot of the 'Fishing Line Status' configuration panel. The title 'Fishing Line Status' is at the top. Below it, there is a section titled 'Line Status' with a downward arrow. This section contains several parameters, each with a corresponding input field or checkbox: 'Max Line Load' (0), 'Over Load Duration' (0), 'Max Line Length' (0), 'Current Line Length' (0), 'Current Over Load' (0), 'Current Line Load' (0), 'Attract Float Speed' (0), 'Is Line Broken' (checkbox), and 'Is Line Breakable' (checkbox with a checkmark).

Fishing Rod Settings is the final element of configuration. **Base Attract Speed** determines the basic speed of attracting the fishing float. Boosts are added to the base speed depending on the tension of the line, the angle between the rod and the float, or the loot tier. **Angle Range** specifies the range of angles for bending the fishing rod.

Fishing Rod Settings

Base Attract Speed5

!

Determines the allowable range of bending angles for the fishing rod. It is used to adjust the bending of the rod based on the calculated angles. The x-component represents the minimum angle, while the y-component represents the maximum angle.

Angle RangeX-110Y110

The configuration of the fishing rod is now complete. You can proceed to test it. By selecting **Show Debug Options**, you will gain access to two variables enabling you to assess the rod's behavior independently of the main Fishing System component.

Show Debug Options

For Debug

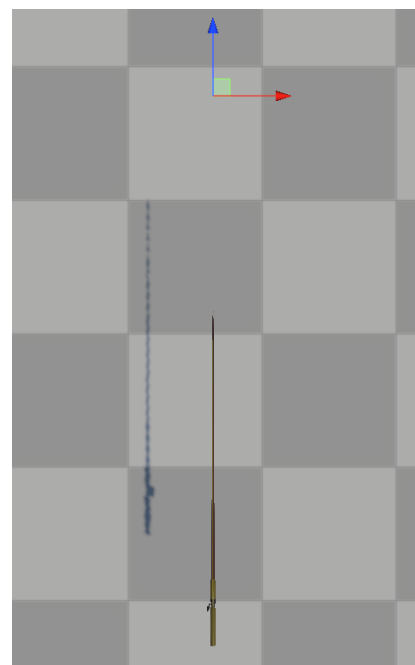
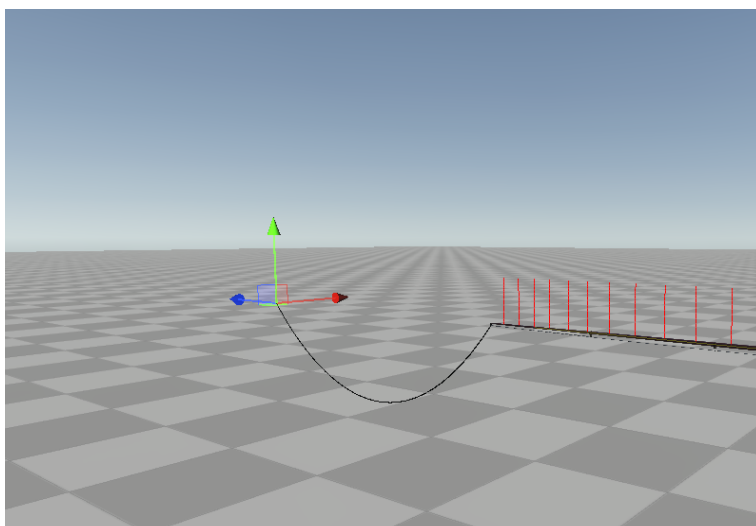
!

The variables below allow you to test the fishing rod during configuration. These variables are modified by the main Fishing System script.

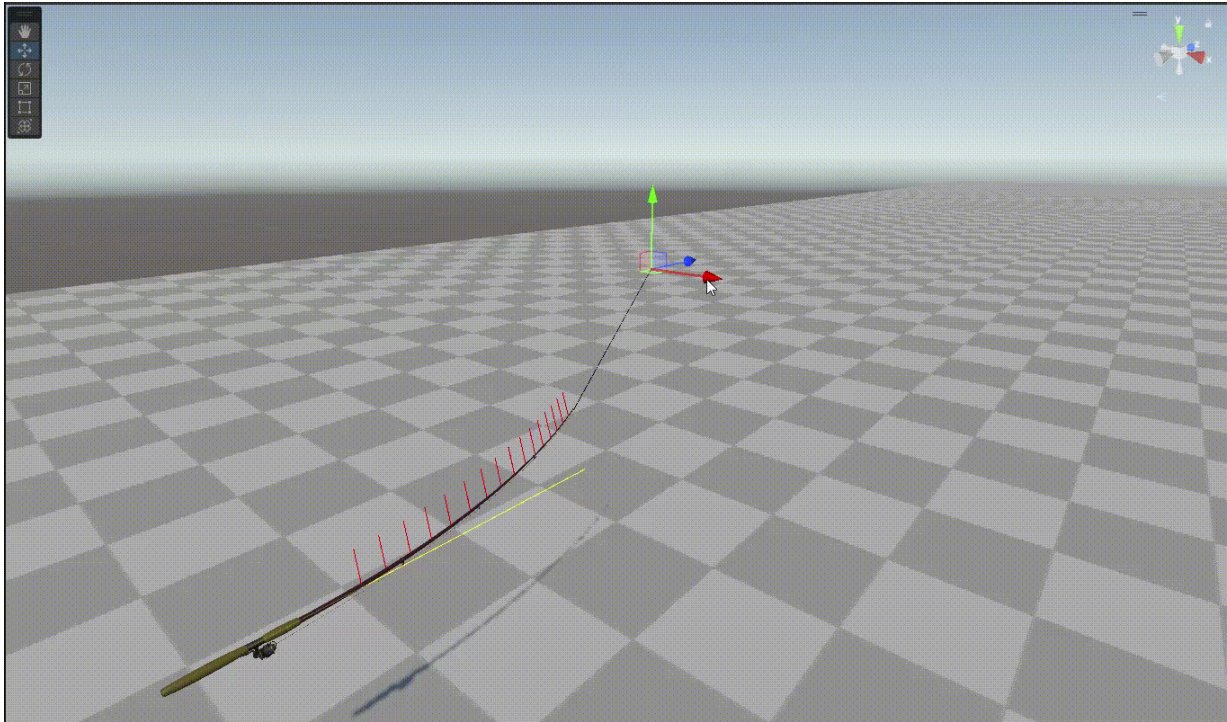
Fishing FloatNone (Transform)

Loot Caught☐

For testing purposes, generate a Game Object within the scene and position it to represent the Fishing Float. In my case, I situated it centrally in front of the fishing rod. This setup will aid in potential Angle Range adjustments. Afterward, press the **Play** button to initiate testing.

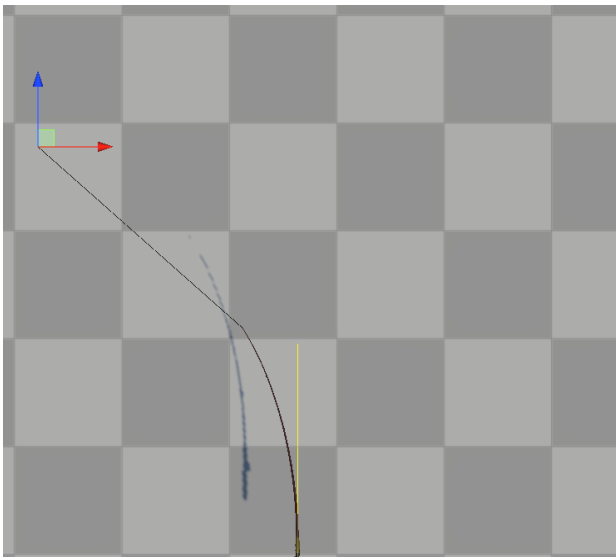


Loot Caught turns on rod bending.

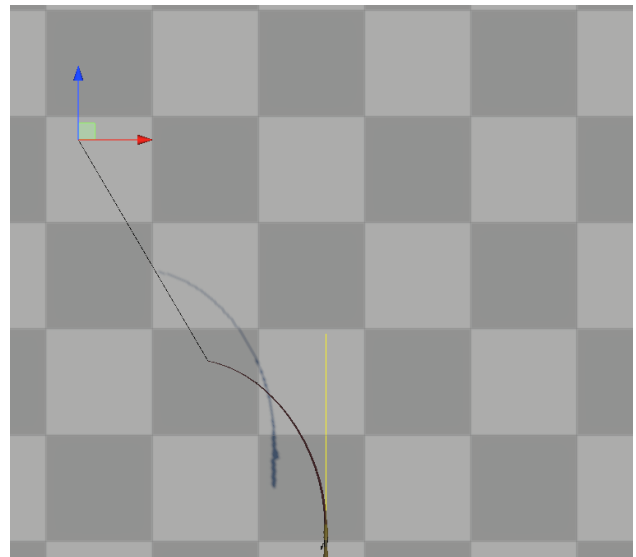


(The gif file in the documentation folder.)

Angle Range Configuration. Incorrect angle may result in the fishing rod not bending precisely towards the fishing float.



(Angle $x = -150$ $y = 150$)



(Angle $x = -60$ $y = 60$)

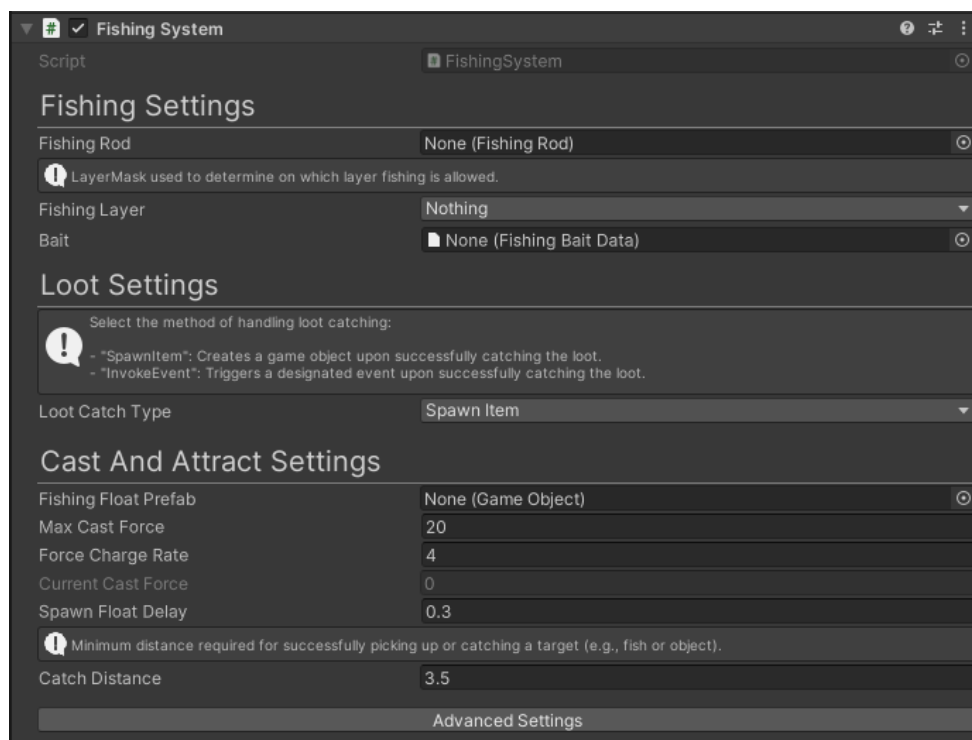
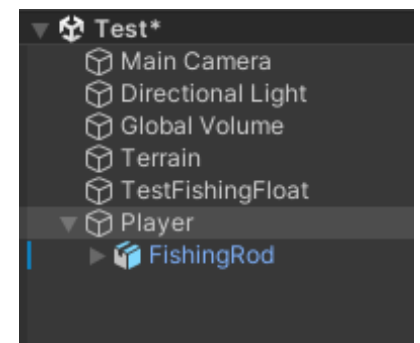
FISHING GAME TOOL

Configure Fishing System

The Fishing System assumes a crucial role as a component engineered to streamline fishing endeavors. Functioning as the core element of your fishing mechanics, this component orchestrates the intricacies of fish-catching activities, encompassing bait management, emulating fish interactions, and bestowing rewards upon successful catches.

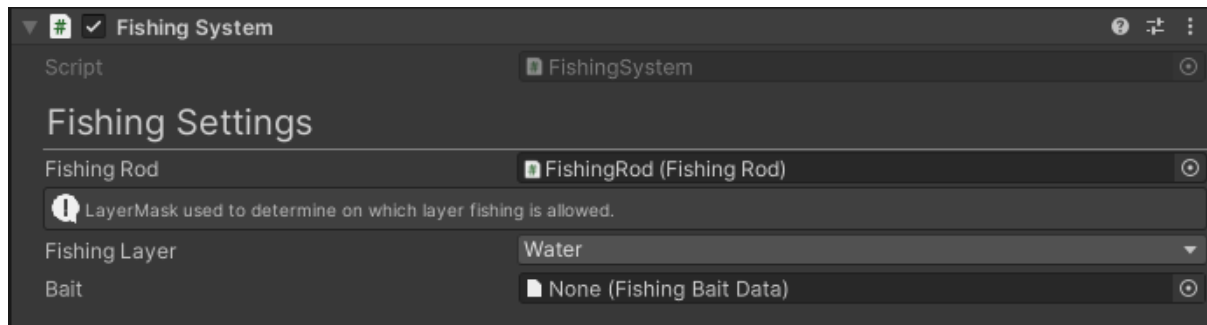
1. Adding the Fishing System Component.

Let's start by adding the Fishing System component to our player. For the purpose of this example, I've created an empty GameObject named Player and placed the previously created fishing rod inside it. To add the Fishing System component, go to **Add Component -> Fishing Game Tool -> Fishing System**.



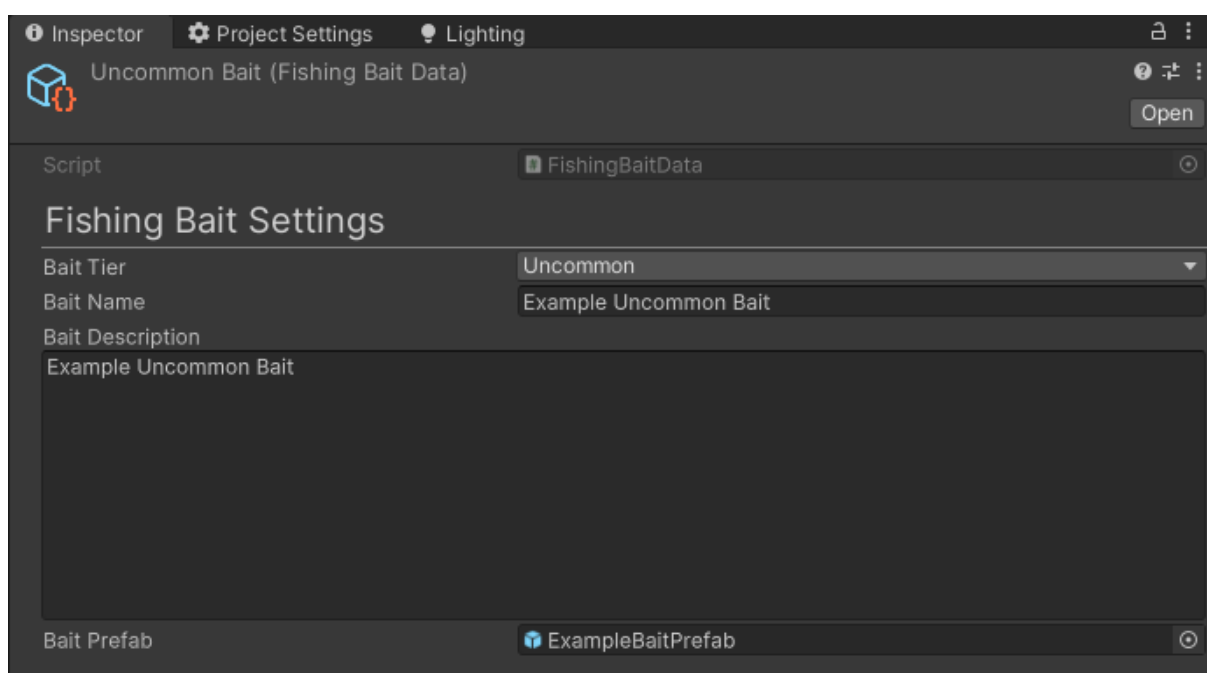
2. Configure Fishing System.

We'll initiate the configuration process from the **Fishing Settings** tab. The **Fishing Rod** parameter accepts values from the currently employed fishing rod. This setup accommodates the inclusion of multiple rods, each with unique parameters.



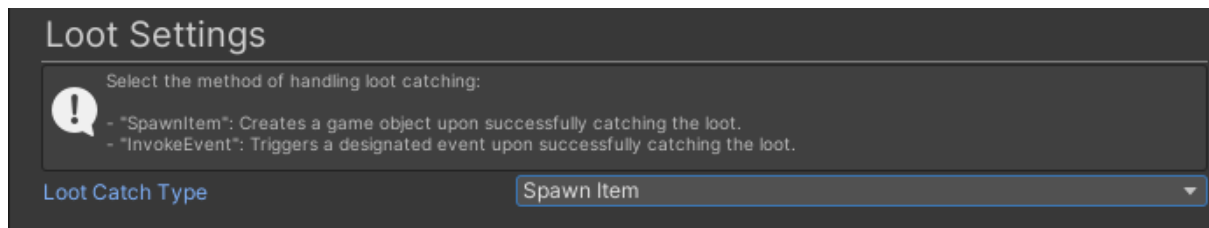
Fishing Layer is used to specify on which layer fishing is allowed. **Bait** parameter takes a scriptable object that specifies the type of bait added. It contains information about the bait's tier, name, description, and bait prefab. The prefab is used when the player wants to add bait after already having added some. The previous bait's prefab spawns in front of the player. The bait's tier determines its quality, which translates into a reduced waiting time for catching a loot on the hook. It also determines the tier of loot that can be caught. For example, using an Epic tier bait allows us to catch loot of a smaller or equal tier, up to Epic tier. Legendary tier loot are excluded.

To add a new bait, you need to click on **Create -> Fishing Game Tool -> New Fishing Bait** in the **Project window**.

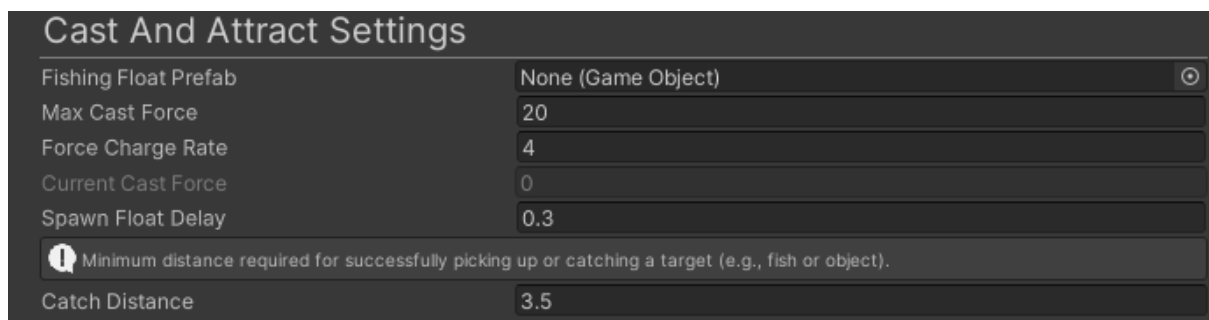


Loot Settings tab is used to select the method of handling the caught loot. **Spawn Item** throws the caught loot onto the shore. **Invoke Event** passes the loot prefab as a GameObject. This means you should add an event that takes a GameObject (public void AddItem(GameObject item)).

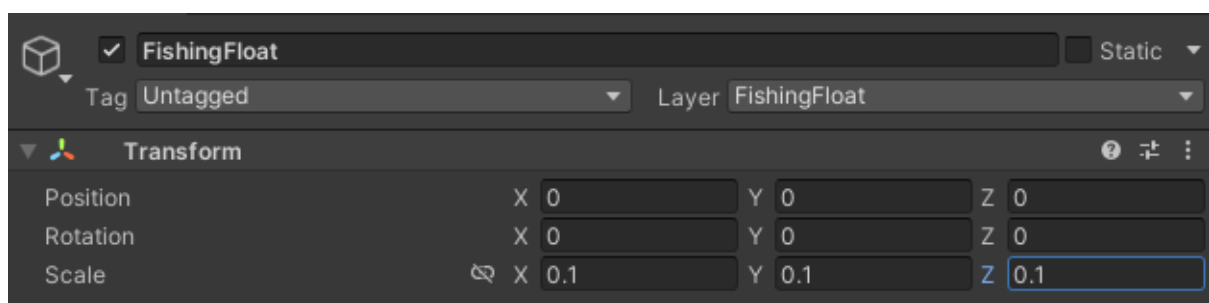
The Invoke Event function is in the development phase and is not functional in version 1.4 of the Fishing Game Tool.



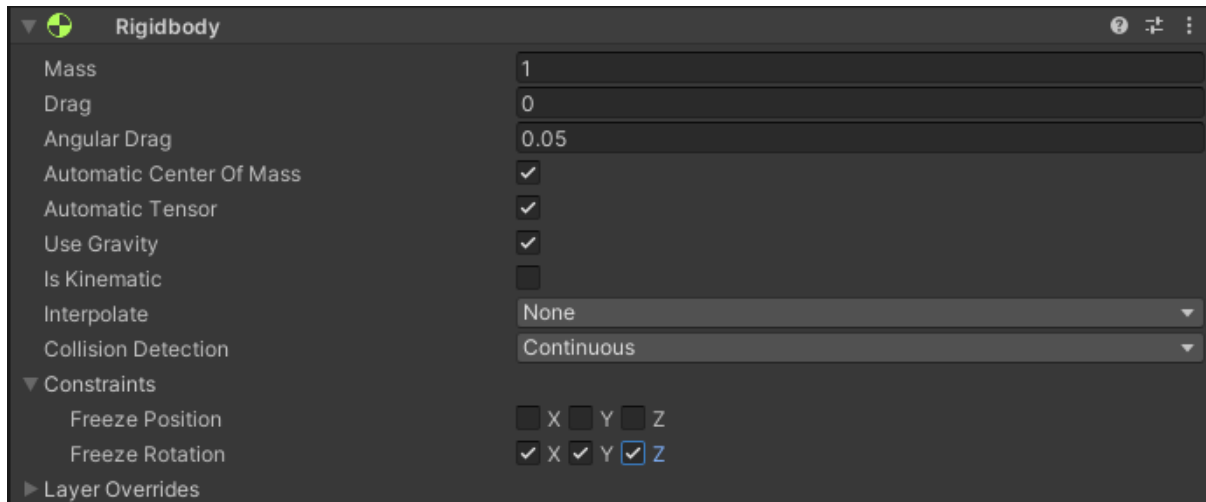
Cast and Attract Settings tab is used for configuring the fishing float.



Fishing Float Prefab stores the fishing float prefab. Let's move on to configuring the fishing float. As an example, I will use a sphere. Set its position to 0,0,0 and scale to 0.1,0.1,0.1. Create a Fishing Float LayerMask and assign it to the float.



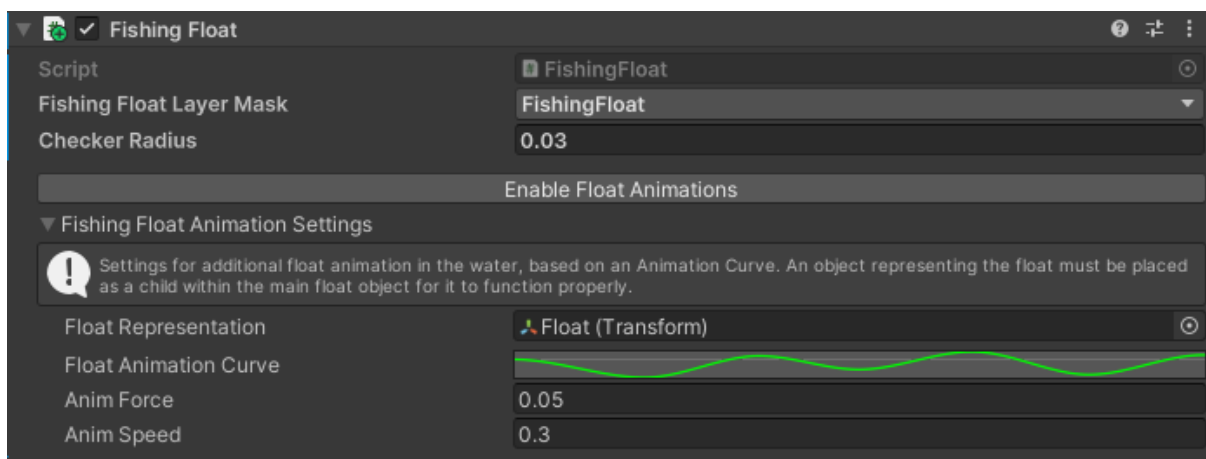
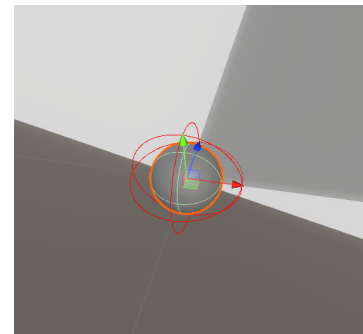
Next, add a Rigidbody. Set Collision Detection to Continuous and lock rotation for the X, Y, and Z axes.



Now, add the Fishing Float component. To do this, go to **Add Component -> Fishing Game Tool -> Fishing Float**.

Set the **Fishing Float Layer Mask** to be the same as that of the float.

Set the **Checker Radius** to extend beyond the outline of the fishing float. You can also add animations to the float. The animation is based on the animation curve so you can easily modify it. To enable the animations press **Enable Float Animations**.

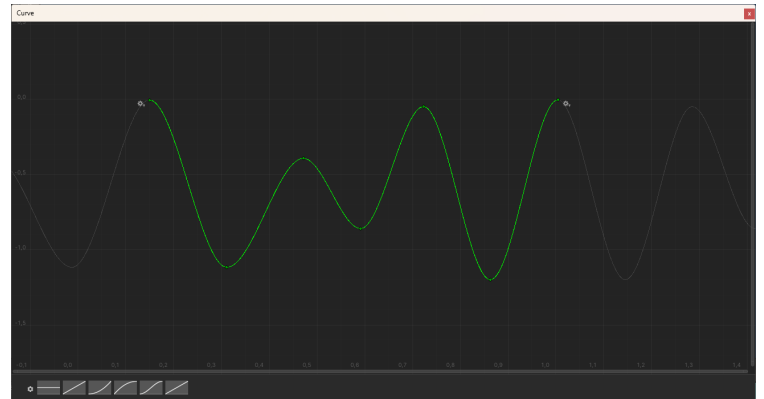


Float Representation is the object representing the float. It must be located in the main float object.



Float Animation Curve

allows you to create your float animation. In my case, the curve takes values from 0 to -1. The right side and the left side have ping pong selected.



Anim Force determines

with what force the animation curve should act on the float. **Anim Speed** determines its speed.

The prepared object should be placed in the Project tab as a prefab. Add the fishing float prefab to the **Fishing Float Prefab** field.

Cast And Attract Settings

Fishing Float Prefab

FishingFloat

Max Cast Force is a parameter that determines the maximum force for casting the fishing float. **Force Charge Rate** defines how quickly the current casting force will increase.

Spawn Float Delay determines the time after which the fishing float will be spawned. This allows you to adjust the timing of the float's release to match the casting animation.

Catch Distance defines the distance between the player and the loot at which the loot is successfully caught.

Cast And Attract Settings

Fishing Float Prefab

FishingFloat

Max Cast Force

20

Force Charge Rate

4

Current Cast Force

0

Spawn Float Delay

0.3

Minimum distance required for successfully picking up or catching a target (e.g., fish or object).

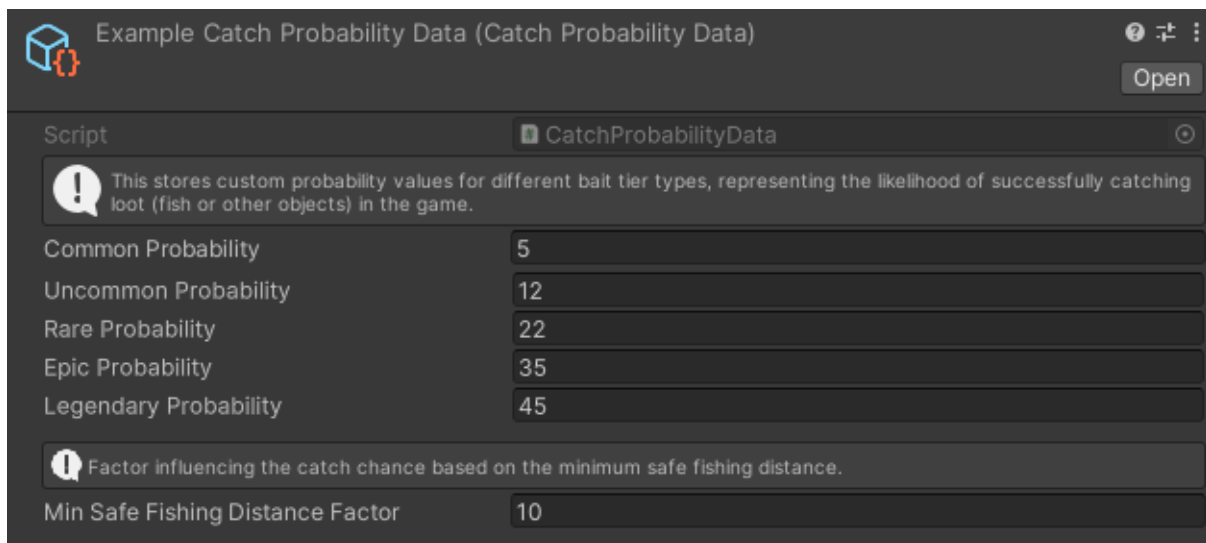
Catch Distance

3.5

The **Advanced Settings** button displays advanced options that can be edited.

The **Catch Probability Data** allows you to modify pre-defined loot catching probability values with your own. To add custom probability data, click on **Create -> Fishing Game Tool -> New Catch Probability Data** in the **Project window**.

Min Safe Fishing Distance Factor determines the minimum distance that does not affect the likelihood of a catch. If the casted float is closer, the probability of catching decreases as the distance decreases.



Example Catch Probability Data (Catch Probability Data)	
Script: CatchProbabilityData	
! This stores custom probability values for different bait tier types, representing the likelihood of successfully catching loot (fish or other objects) in the game.	
Common Probability	5
Uncommon Probability	12
Rare Probability	22
Epic Probability	35
Legendary Probability	45
! Factor influencing the catch chance based on the minimum safe fishing distance.	
Min Safe Fishing Distance Factor	10

Caught Loot allows you to force the capture of loot. It is used for testing purposes.

Return Speed Without Loot determines the speed at which the fishing float is reeled in when no loot has been caught on the hook.

Catch Check Interval determines how frequently the check is made to see if the loot has been caught on the hook. The values are in seconds.

The fishing system has been successfully configured! Now let's move on to configure the rest of the components.

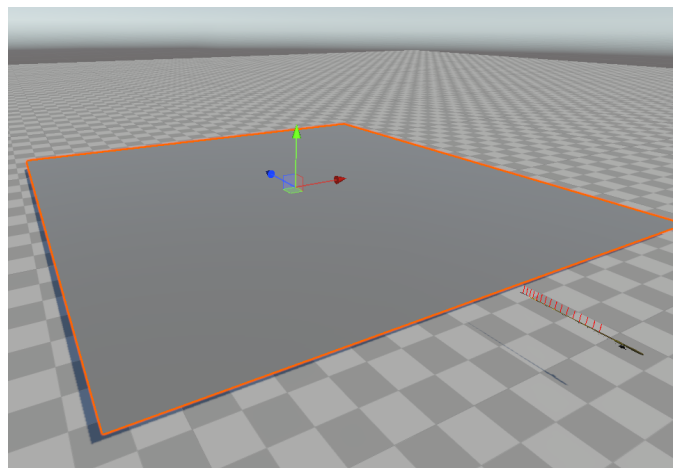
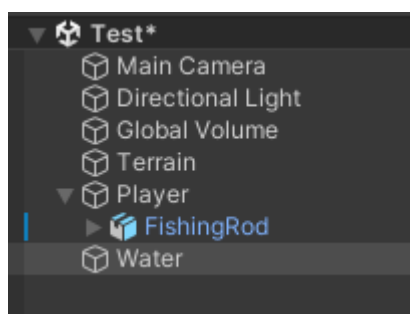
FISHING GAME TOOL

Configure The Fishing Spot

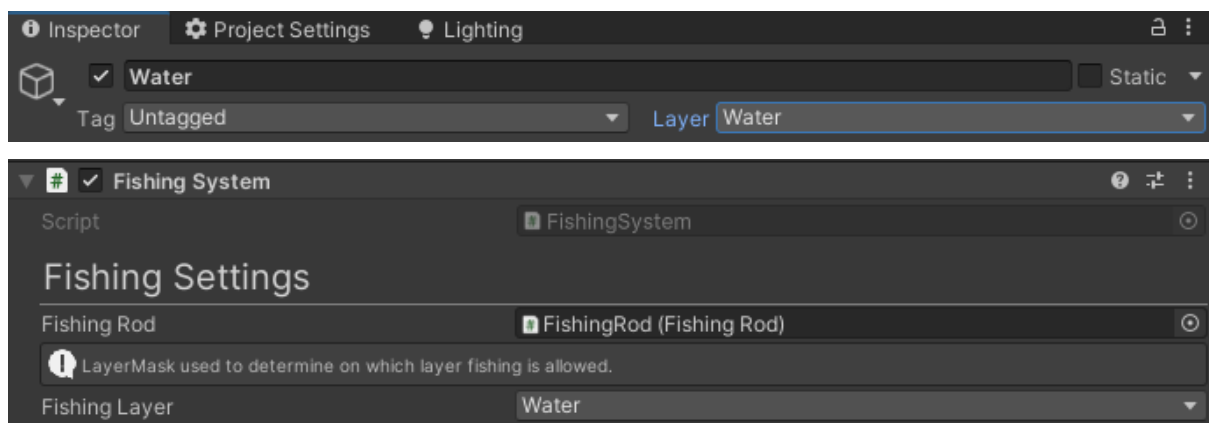
The Fishing Game Tool facilitates the setup of multiple distinct fishing spots, each capable of featuring its own exclusive loot for acquisition. The shape or size of the fishing spot does not impose limitations on this capability.

1. Create a fishing surface.

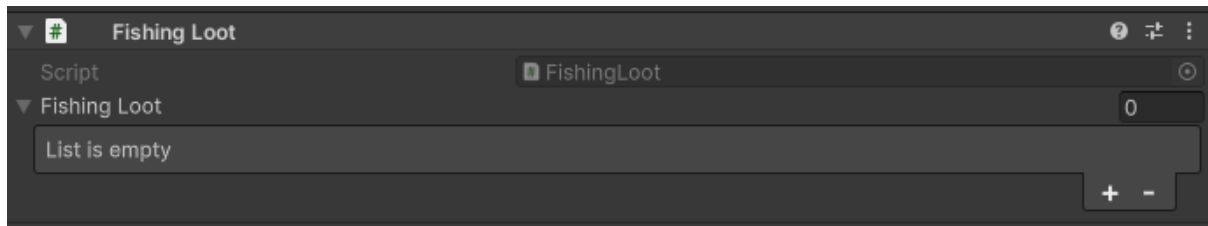
As an example, I used a Plane, which will represent the fishing surface.



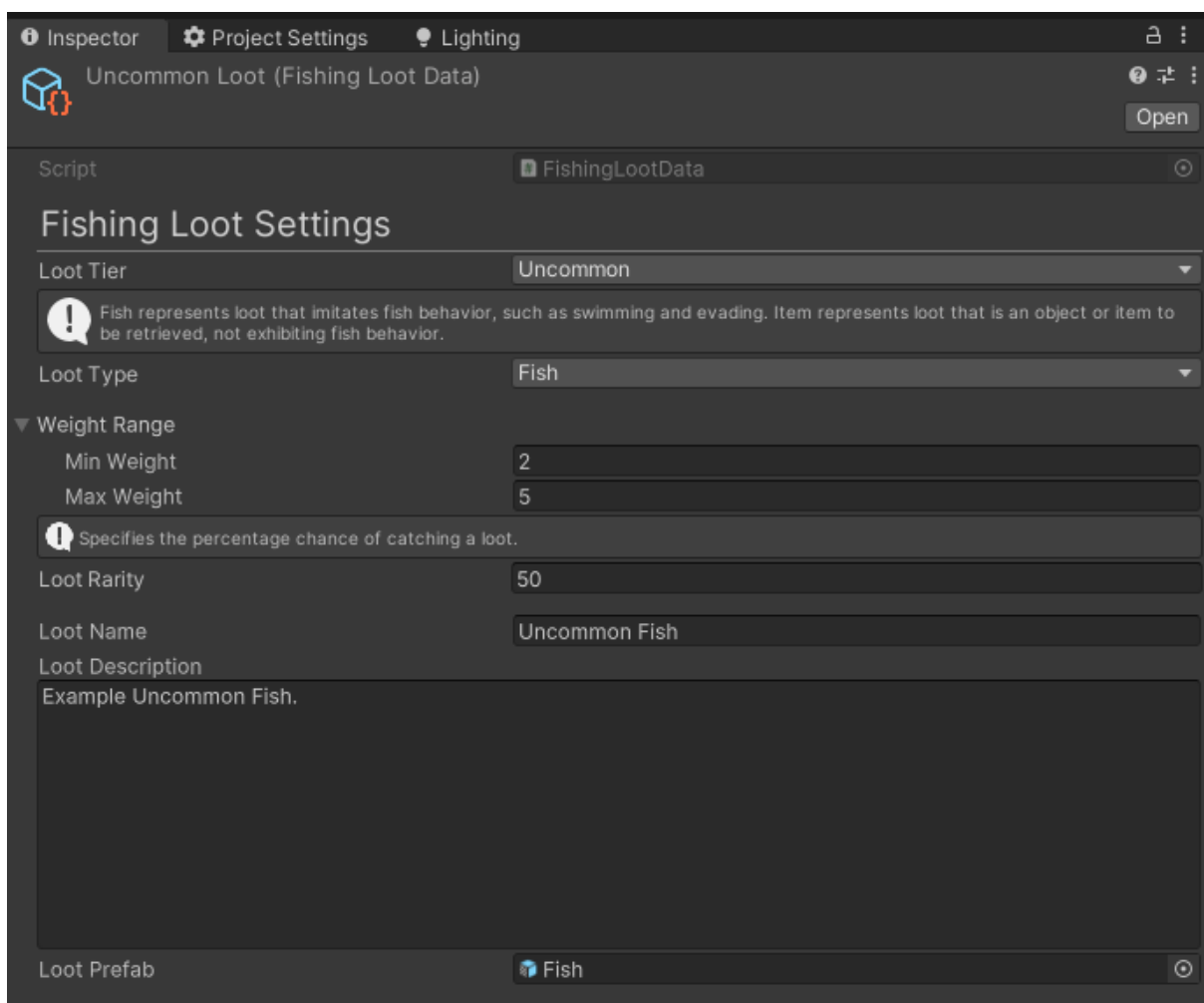
The next step will be setting the Layer for the created water. The Layer must be the same as the one set in the **Fishing Layer** in the **Fishing System**.



Let's add **Fishing Loot** to the water object. To do this, click on **Add Component -> Fishing Game Tool -> Fishing Loot**.



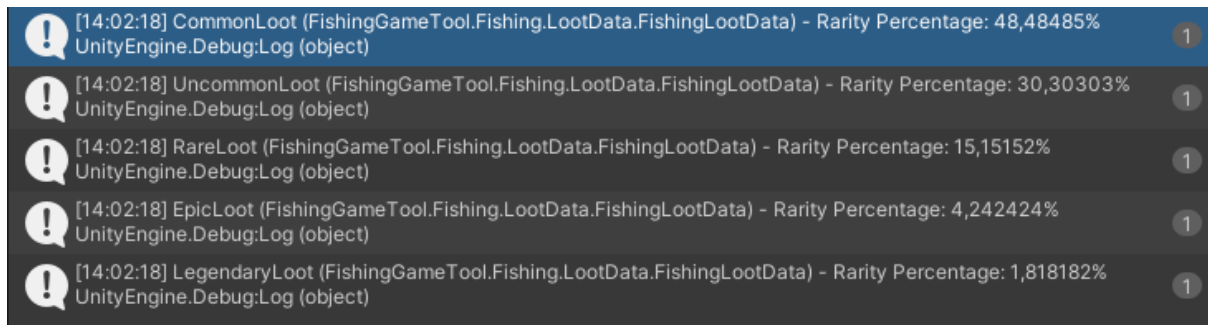
Fishing Loot stores the available loot for a given fishing area. To create loot, click on **Create -> Fishing Game Tool -> New Fishing Loot** in the **Project window**.



Loot Type determines the type of loot, which can be either Fish or Item.

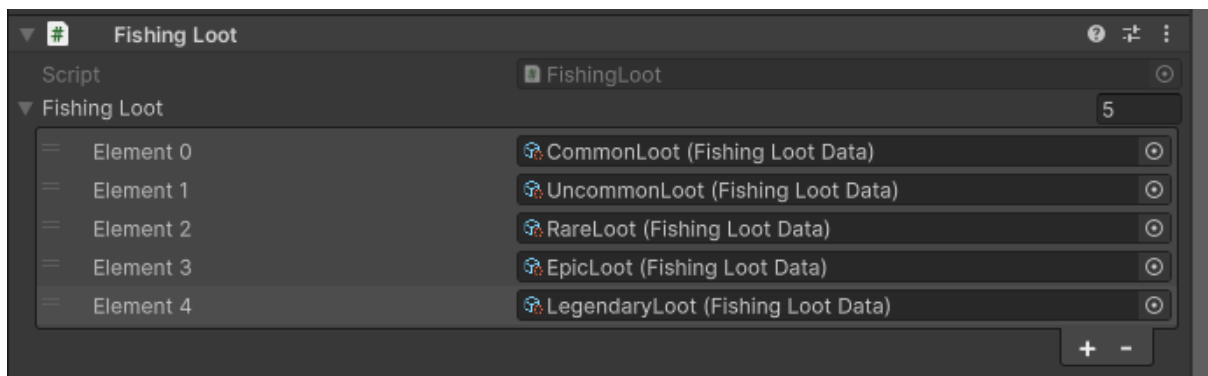
Weight Range stores the weight range for a specific loot. The final weight is randomly generated within the range of the minimum and maximum values of the variable.

Loot Rarity determines the rarity of the loot. The values of all available loot for a specific fishing spot are calculated and scaled to a range of 0-100. There's no need to calculate them manually.



Loot Prefab stores the loot prefab. This prefab will be spawned when the loot is caught.

For testing purposes, I created five different types of loot, ranging from common to legendary. Subsequently, I placed them in the **Fishing Loot** component.



We have just configured the **Fishing Game Tool**. The fishing system is now operational and can be tested. The right mouse button is responsible for reeling in the fishing float, while the left mouse button is used for casting it.

FISHING GAME TOOL

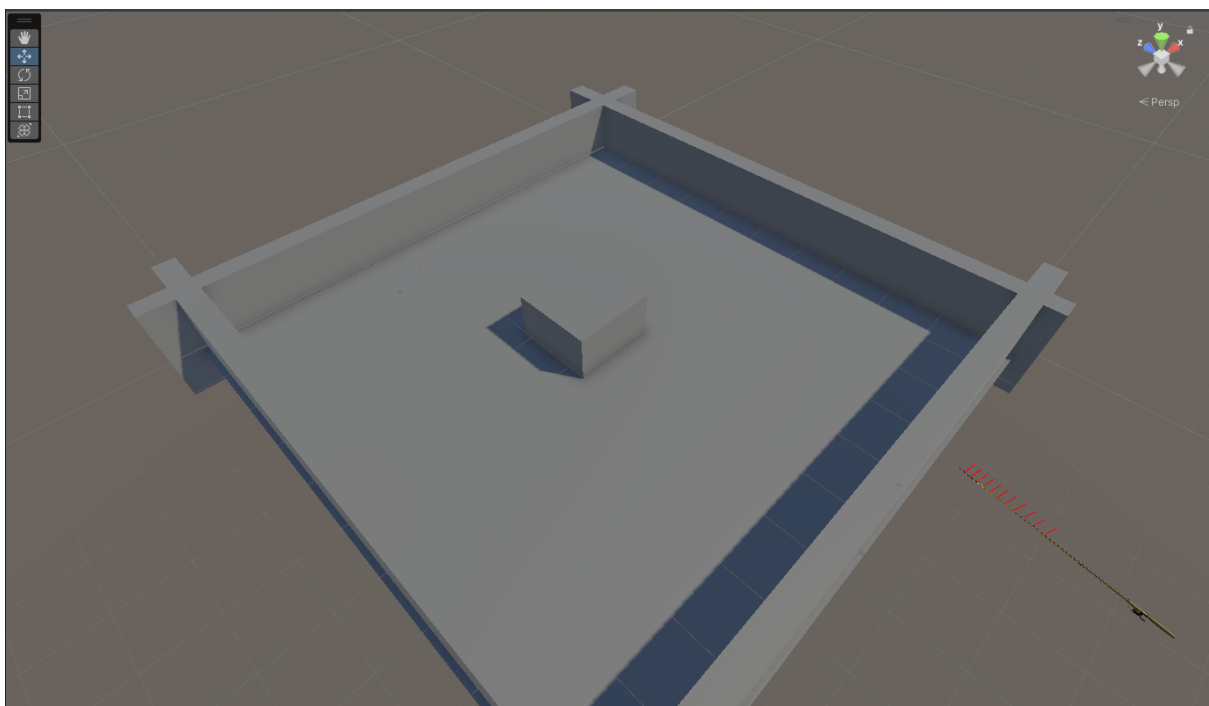
Testing

The configuration of the Fishing Game Tool has been completed. You can now advance to the testing phase and evaluate the functionality of all subsystems. This testing process will help ensure that all components interact as expected and contribute to a seamless fishing experience within your project.

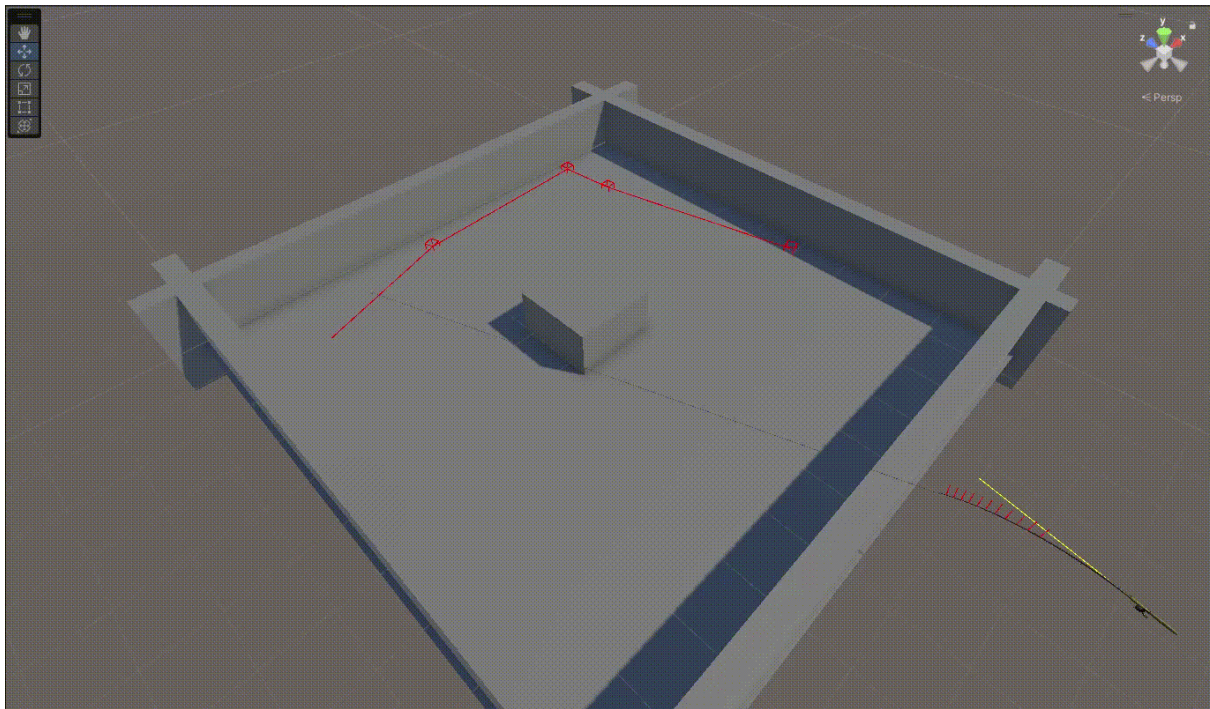
1. Simulation of fish behavior.

The Fishing Game Tool has been augmented with a fish behavior simulation feature. This advancement equips it with the ability to identify shorelines and other obstacles within the water, adapting its navigation to circumvent them. The tool adjusts its direction and speed, infusing the simulation with a heightened sense of realism. This enhancement ensures a more immersive fishing experience within your project.

To test this, I added a shoreline and placed an obstacle in the middle.



Press Play!

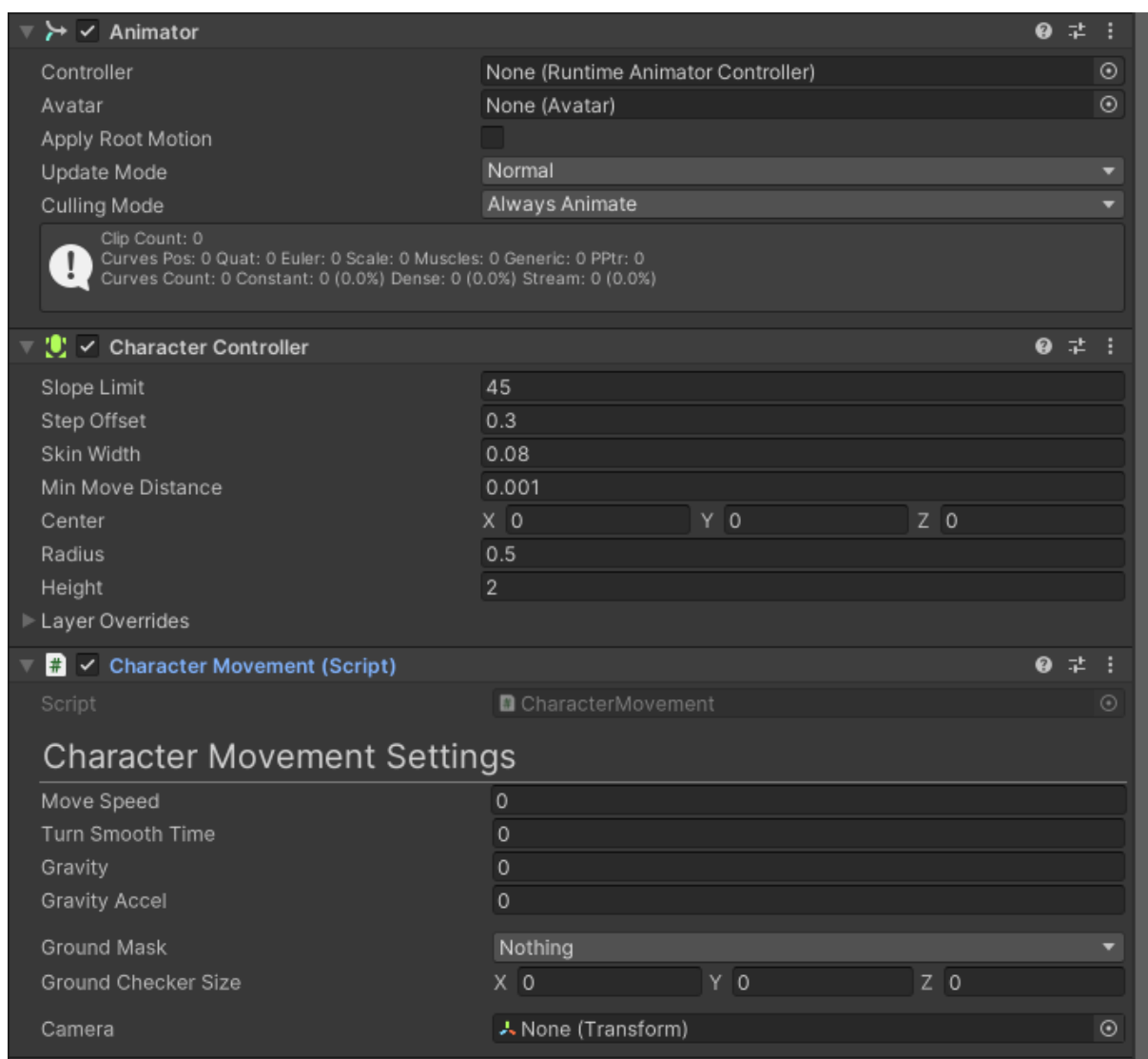


(The gif file in the documentation folder.)

FISHING GAME TOOL

Character Movement

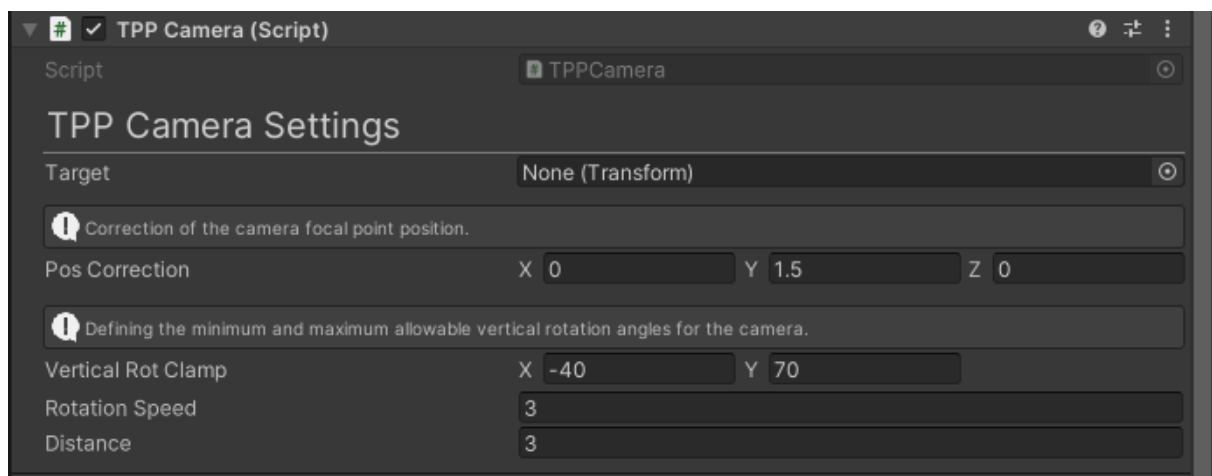
Character Movement is a straightforward script designed to incorporate the capability for character mobility within the project.



FISHING GAME TOOL

TPP Camera

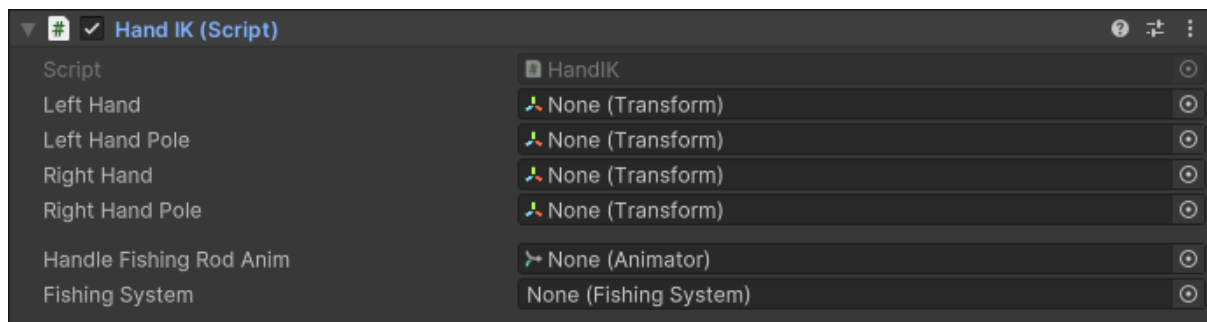
TPP Camera is a script that adds the ability to control the camera from a third-person perspective.



FISHING GAME TOOL

Hand IK

Hand IK is a basic script designed to oversee a character's hand movements. In the provided scene, where the **Fishing Rod Handle** is animated, **Hand IK** facilitates the synchronization of the character's hands with the movement of the fishing rod. This script ensures that the hands remain appropriately positioned in relation to the fishing rod, contributing to a realistic portrayal of the character's interaction with the rod.



FISHING GAME TOOL

Simple UI Manager

Simple UI Manager demonstrates how to construct a user interface (UI) for the established system. It offers the ability to customize colors and fill direction for a progress bar, as well as the option to activate a gradient color effect that corresponds to the progress of the bar. This script exemplifies the process of creating an engaging and interactive UI element within the system.

