Bow and Arrow Controller - Documentation

Welcome to the **Bow and Arrow Controller** documentation! This guide is designed to help developers and buyers understand the features, setup, and usage of the Bow and Arrow Controller Unity asset. Whether you're integrating the controller into your game or customizing it for your needs, this documentation will serve as your starting point.

Table of Contents

---

1. Overview

The **Bow and Arrow Controller** is a highly customizable Unity asset designed to simulate realistic bow and arrow mechanics. It includes features such as string tension, arrow spawning, multiple shooting patterns, and Inverse Kinematics (IK) integration for character animation. The asset comes with a fully configured player prefab, making it easy to integrate into your project.

---

2. Features

- **Realistic Bow Mechanics**: Simulates string tension, bow limb bending, and recoil effects.
- **Customizable Arrow Patterns**: Shoot arrows in various patterns (linear, circle, star, etc.).
- **Multiple Shooting Types**: Parallel, sequential, random, alternating, and more.
- **Inverse Kinematics (IK)**: Supports hand positioning for character animation.
- **Visual Effects**: Emission color customization for the bow and arrows.
- **Player Prefab**: A ready-to-use player setup with animations and controls.

---

3. Setup

Importing the Asset

1. Download and import the **Bow and Arrow Controller** package into your Unity project.
2. Ensure that the required dependencies (e.g., Unity's Cinemachine, Input System) are installed.

Player Prefab Setup

1. Locate the `PlayerPrefab` in the `Prefabs` folder.
2. Drag and drop the `PlayerPrefab` into your scene.
3. Configure the player's input settings (e.g., mouse click for shooting) in the `Input Manager`.

Bow Configuration

1. Create a new `BowConfig` ScriptableObject by navigating to `Assets > Create > ScriptableObjects > BowConfig`.

2. Customize the bow's settings (e.g., arrow speed, string tension, emission color) in the `BowConfig` asset.
3. Assign the `BowConfig` asset to the `Bow` script in the `PlayerPrefab`.

---

4. BowConfig ScriptableObject

The `BowConfig` ScriptableObject is the central configuration file for the bow. It allows you to customize various aspects of the bow's behavior.

Bow References

- **Arrow Prefab**: The arrow GameObject to be shot.
- **Bow Emission Color**: The emission color for the bow's material.

Arrow Settings

- **Arrow Spawn Offset**: Local position offset for arrow spawn point.
- **Arrow Spawn Rotation**: Initial rotation of spawned arrows.
- **Arrow Initial Local Position Z**: Initial Z position of the arrow when nocked.

Bow Physics

- **Bow Spring Constant**: Spring force constant for string tension.
- **Bow Damping Factor**: Damping factor for string movement.
- **Use Gravity**: Whether the bow is affected by gravity.

Bow Settings

- **Bow Idle Position**: Bow position when not drawn.
- **Bow Idle Rotation Z**: Z-axis rotation when the bow is at rest.
- **Bow Draw Position**: Bow position when fully drawn.
- **Bow Draw Rotation Z**: Z-axis rotation when the bow is fully drawn.

String Mechanics

- **Max String Pull Distance**: Maximum distance the string can be pulled back.
- **String Initial Y Position**: Initial Y position of the string endpoints.
- **String Final Y Position**: Final Y position of the string when drawn.

Shooting Settings

- **Arrow Launch Speed**: Base speed of launched arrows.
- **Number of Arrows**: Number of arrows to shoot simultaneously.
- **Angle Between Arrows**: Angle between multiple arrows (in degrees).

- **Arrow Pattern**: Pattern for arranging multiple arrows (e.g., linear, circle).
- **Shooting Type**: Firing sequence type (e.g., parallel, sequential).

IK Settings

- **Update Left Hand IK Position**: Whether to update the left hand's IK position.
- **Left Hand IK Position Offset**: Offset for the left hand's IK position.

---

5. Bow Script

The `Bow` script handles the core functionality of the bow, including string mechanics, arrow spawning, and shooting logic.

Key Components

- **BowConfig**: Reference to the `BowConfig` ScriptableObject.
- **Arrow Spawn Point**: Transform where arrows are spawned.
- **Right/Left Hand Targets**: IK targets for character animation.
- **Line Renderer**: Visual representation of the bowstring.

Initialization

- The bow's settings are initialized in the `Start` and `OnEnable` methods.
- The bow's emission color and string position are set based on the `BowConfig`.

String Mechanics

- The string's position is updated based on player input (mouse click).
- String tension and damping are simulated using a spring-damper system.

Arrow Management

- Arrows are spawned and managed in a list.
- Arrow positions are updated based on string displacement.

Shooting Logic

- Arrows are shot based on the selected shooting type (e.g., parallel, sequential).
- The shooting logic is handled in the `ShootArrowsRoutine` coroutine.

---

6. Player Prefab

The `PlayerPrefab` includes a complete setup for the bow and arrow controller, including:

- **Animator**: Handles character animations.
- **IK Targets**: For realistic hand positioning.
- **Camera**: Follows the player and aims the bow.

---

7. Customization

Adjusting Bow Behavior

- Modify the `BowConfig` settings to change the bow's physics, appearance, and shooting behavior.

Adding New Arrow Patterns

- Extend the `ShapePattern` enum and update the `SpawnArrows` method to support new patterns.

Modifying Shooting Types

- Add new shooting types by extending the `ShootingType` enum and updating the `ShootArrowsRoutine` method.

---

8. Troubleshooting

- **Issue**: Arrows not spawning.
  - **Solution**: Ensure the `Arrow Prefab` is assigned in the `BowConfig`.
- **Issue**: Bowstring not moving.
  - **Solution**: Check the `Line Renderer` component and `BowConfig` string settings.

---

9. Support

For additional support, please contact us at ragerungames@gmail or DM me on discord https://discord.gg/ZHxB9MkkR6

Thank you for using the **Bow and Arrow Controller**! We hope this documentation helps you create amazing archery experiences in your Unity projects. Happy developing! 🎯