### **Documentation: Kick Module Client**

This module handles the client-side logic for a football kicking mechanic in Roblox, including the GUI, power, accuracy, and angle controls for the kick. It also manages camera transitions during the kick process.

#### **Dependencies and Services**

* **players**: Fetches the Players service to interact with the local player.
* **tweenService**: Manages GUI animations, specifically the kick meter.
* **replicatedStorage**: Handles communication between the client and server through remotes.
* **kickGui**: References the GUI elements for the kick, specifically the KickGui which displays the kick meter and other controls.
* **playerControls**: Provides control over the player's movements, enabling and disabling controls during the kicking process.
* **currentCamera**: Controls the camera view for the kicking process.

#### **Key Variables**

* **power**: Stores the power level of the kick based on the player's selection.
* **accuracy**: Stores the accuracy value selected by the player.
* **verticalAngle**: Represents the vertical angle of the kick, ranging from 15° to 55°.
* **kickMeterForwardTween** and **kickMeterBackwardTween**: Tween animations for the kick meter's selector, which moves back and forth during the power/accuracy selection.

#### **Functions**

1. **kickModuleClient:StartKicking(kickSetup)**:
   * Disables player controls and switches the camera to a scriptable camera that focuses on the CameraPart provided in kickSetup.
   * Activates the kick meter GUI and starts the forward and backward tween animations to allow power and accuracy selection.
2. **kickModuleClient:DetermineSelection()**:
   * Returns which value (Power or Accuracy) is currently being selected based on the state of power and accuracy.
3. **kickModuleClient:ResetGui()**:
   * Resets the kick meter to its default position and cancels any active tween animations.
   * After a short delay, it restarts the forward movement of the selector.
4. **kickModuleClient:SelectPower()**:
   * Calculates and sets the power value based on the position of the kick meter selector.
   * Resets the GUI and moves on to accuracy selection.
5. **kickModuleClient:SelectAccuracy(kickSetup)**:
   * Calculates and sets the accuracy value.
   * Resets the GUI and proceeds to finalize the kick by calling CompleteKick.
6. **kickModuleClient:AdjustAngleUp(kickSetup)**:
   * Increases the vertical angle of the kick in small increments (capped at 55°) and updates the orientation of the KickArrow.
7. **kickModuleClient:AdjustAngleDown(kickSetup)**:
   * Decreases the vertical angle of the kick in small increments (no lower than 15°) and updates the KickArrow.
8. **kickModuleClient:CompleteKick(kickSetup)**:
   * Finalizes the kick by hiding the GUI, re-enabling player controls, and returning the camera to follow mode.
   * Fires a remote event to the server, sending the final values for power, verticalAngle, and accuracy.