PROJECT DOCUMENTATION

Basic Third Person Controller

Version: 1.0.0

Author/Studio: Rafael Damasceno Maques dos Santos/ Pbwanskaas Productions

Contact: rafaeldms0611@gmail.com

1. Project Overview

This asset is designed to streamline game development by offering a complete and efficient third-person movement solution. It features an integrated foot placement system, a pre-set Animator, and a dynamic focus-style camera, enabling developers to enhance their projects and avoid the need to build core mechanics from scratch.

2. Getting Started & Installation

Prerequisites:

- Cinemachine. Animation Rigging. Input System

Installation Steps:

1. **Import** the project and its **dependencies**.
2. You can use the template **prefabs** as shown in the test scene, or create from scratch.
3. **Changing the model**: In the player hierarchy, replace the **Y-bot model** with your own. Remember to keep the **IK rig** and just transfer it to your model. On your model, add **foot placement**, **rig builder**, and the **animator**.
4. **Changing animations**: For each animation, you'll need to use record mode to change the **Weight** values of the **Two Bone IK constraint** (these are found in each of the **footIK** components).
5. Position the **kneeIK** objects on their respective knees. If necessary, adjust them according to the animation.
6. The **foot placement** system only works for inclines. You can disable it if you prefer by removing all **Rig builder** components, as well as the **foot placement** script.
7. **IMPORTANT**: Keep your animations and model in **Generic** format. An avatar is not required.

3. Folder Structure

- **/Animations:**
  - **/Player:** All player animations, as well as their controller. **IMPORTANT**: All rights to the animations and model are reserved for mixamo.com. The animations and model were used from there and are **not for commercial use**. The project's focus is on the movement mechanics.
- **/Materials:** All materials, and Icon
- **/Prefabs:** Contains ready-to-use prefabs like Player, Enemy, Main Camera, elements.]
- **/Models:**
  - **/TestModels:** the Y-Bot model, from Mixamo
- **/Scripts:** The player, and important scripts
  - **/Enemy:** The simple movement enemy script
- **/Scenes:** The test scene, with all the prefabs working
- **/Shaders:** the shader from Target Icon

4. Player Prefab

Key Scripts:

Script Name	Description
ThirdPersonController.cs	Handles all player input for movement
FootPlacement.cs	Control the basic Foot Placemet System
StarterAssetsInputs.cs	Control the game inputs. Put here your new inputs

Customization (Inspector Fields for ThirdPersonController.cs):

Field Name	Data Type	Description
Move Speed	float	The speed at which the player walks. Default: 5.335
Sprint Speed	float	The speed at which the player runs. Default: 10
Targeting Move Speed	float	The speed at which the player walks in foccus. Default: 1.5

Rotation Smooth Time	slider	How fast the character turns to face movement direction. Default: 0.12
Speed Change Rate	float	Acceleration and deceleration. Default: 20
Sticking Force	float	Force that pulls the character to the ground to prevent jumping on ramps. Default: 10
Grounded Offset	float	The offset of the ground check sphere. Default: -0.14
Grounded Radius	float	The radius of the ground check sphere. Default: 0.28
Ground Layers Ground Inclined	LayerMask	What layers the character uses as ground
ft	FootPlacement	your obj with footplacement script
Rbuild	Rig Builder	your obj with Rig Builder component
Rig Player	Game Object with tag “MyRig”	Your IK Rig
Obstacle Layers	LayerMask	Set the layers that should be considered obstacles to block the aim (walls, etc.).
Max Target Distance	float	Distance to Stay in Focus Mode. Default: 16
TargetIconHeight	float	the height of your target icon
TargetRotationSmoothTime	float	How smoothly the player rotates when focusing on a target. Lower = faster.
Target Aim Offset	Vector3	Vertical and horizontal adjustment of the aiming point on the target (to aim at the chest instead of the feet).

CinemachineCameraTarget	Game Object	The follow target set in the Cinemachine Virtual Camera that the camera will follow
Camera Target Offset	Vector3	Offset for the point the camera will aim at, from the player's pivot. Adjust the Y to aim at the chest/head.
TopClamp	float	How far in degrees can you move the camera up. Default: 50
Bottom Clamp	float	How far in degrees can you move the camera down. Default: -15
Camera Angle Override	float	Additional degress to override the camera. Useful for fine tuning camera position when locked. Default: 0
Lock Camera Position	bool	For locking the camera position on all axis
Camera Sensitivity	float	Camera sensitivity. Works for mouse and controller. 0.2 it's a good number for controllers. try 0.5 for mouse

FootPlacement Script

Customization (Inspector Fields):

Field Name	**Data Type**	**Description**
left Foot Bone right Foot Bone	transform	"Bone References (Drag from Skeleton)
left Foot IK Target right Foot IK Target	transform	IK Targets
left Foot IK Constraint right Foot IK Constraint	TwoBoneIKConstraint	IK Constraints
layer Mask	LayerMask	your raycast to check ground

distance To Ground raycast Distance	float float	
rotation Offset	Vector3	Use this to fine-tune the foot rotation.

5. Frequently Asked Questions (FAQ)

Q: Can I use my own 3D models for the player/enemies?
A: Yeah, just follow the step-by-step in 2.3
Q: I'm getting an error about 'X' package missing.
A: This project requires the “Cinemachine”, “Animation Rigging” and “Input Manager” packages. Please install it from the Unity Package Manager by going to Window > Package Manager, finding the packeges, and clicking 'Install'

7. Support & Changelog

Support:
If you have any issues or questions, please contact us at:
rafaeldms06@gmail.com
We will do our best to assist you!

Changelog:

- **v1.0.0 (27/06/2025):**
  - Initial release.

8. License & Terms of Use

This asset is licensed under the Standard Unity Asset Store End User License Agreement (EULA). For full details on the terms of use, please refer to the official Unity Asset Store documentation.

You are **allowed** to:

- Use and modify the asset for your games, animations, and other creative projects.
- Release commercial projects that use this asset.

You are **NOT allowed** to:

- Resell, redistribute, or share the asset, in its original or modified form, as a standalone product.
- Claim ownership of the asset.