MANUAL

DOCUMENTATION

Kreshnik Halili, MSc

HOAX GAMES
SOCIAL TRUTH FIGHTER

TABLE OF CONTENTS

HOAX GAMES

SOCIAL TRUTH FIGHTER

INSTALLATION
URP DEMO

To get started with the UPR demo scene you have first to import the corresponding package.

From the folder "Assets_HoaxGames\iStep\" import the package:

"iStep_Demo_URP.unitypackage"

REQUIRED PACKAGES

The demo scene requires the following Unity packages to be installed:

- *Cinemachine*
- *Input System*

Typically, the within the package included "Starter Assets" subset will install those dependencies automatically through the included Editor scripts. Should there be any issues please try to resolve them by going to "Tools" → "Starter Assets" → "Reinstall Dependencies".

Alternatively, you can install the required packages (dependencies) manually using the “Package Manager”. To do so go to “Window” → “Package Manager” and change the packages to show settings to “Packages: Unity Registry” in the top left corner of the “Package Manager” window (the setting is a drop-down menu).

Select “Cinemachine” from the list and press “install” (bottom right corner).

Select “Input System” from the list and press “install” (bottom right corner).

If you went for the manual installation, please make sure that under “Edit” → “Project Settings” → “Player” → “Other Settings” you have “STARTER_ASSETS_PACKAGES_CHECKED” added to the “Scripting Define Symbols”.

ANIMATOR SETTINGS

iStep requires an IK pass to be enabled in the Animator. To do so open the imported demo scene (“iStepShowcaseDemoURP.unity”) and select the gameobject “PlayerArmature” from the hierarchy. From the “Animator” component attached to “PlayerArmature” double-click the object assigned under the “Controller” variable (this is the corresponding “AnimatorController” object called “StarterAssetsThirdPerson”). This should open the players “AnimatorController” in the “Animator” window. Alternatively, you can search for “StarterAssetsThirdPerson” in the “Project” tab and double-click the found “AnimatorController”.

In the “Animator” window press the settings icon next to “Base Layer” under the “Layers” tab and activate the “IK pass” for the “Base Layer”.

URP SETTINGS

To make sure everything works flawlessly (like for example the footstep decals) one last step is required.

Go to “Edit” → “Player Settings” → “Quality” and set the “Render Pipeline Asset” to “UniversalRP-HighQuality” from “Assets_HoaxGames\iStep_Demo_URP\Settings\”.

INSTALLATION
HDRP DEMO

To get started with the HDPR demo scene you have first to import the corresponding package.

From the folder “Assets_HoaxGames\iStep\” import the package:

“iStep_Demo_HDRP.unitypackage”

REQUIRED PACKAGES

The demo scene requires the following Unity packages to be installed:

- *Cinemachine*
- *Input System*

Typically, the within the package included “Starter Assets” subset will install those dependencies automatically through the included Editor scripts. Should there be any issues please try to resolve them by going to “Tools” → “Starter Assets” → “Reinstall Dependencies”.

Alternatively, you can install the required packages (dependencies) manually using the “Package Manager”. To do so go to “Window” → “Package Manager” and change the packages to show settings to “Packages: Unity Registry” in the top left corner of the “Package Manager” window (the setting is a drop-down menu).

Select “Cinemachine” from the list and press “install” (bottom right corner).

Select “Input System” from the list and press “install” (bottom right corner).

If you went for the manual installation, please make sure that under “Edit” → “Project Settings” → “Player” → “Other Settings” you have “STARTER_ASSETS_PACKAGES_CHECKED” added to the “Scripting Define Symbols”.

ANIMATOR SETTINGS

iStep requires an IK pass to be enabled in the Animator. To do so open the imported demo scene (“iStepShowcaseDemoHDRP.unity”) and select the gameobject “PlayerArmature” from the hierarchy. From the “Animator” component attached to “PlayerArmature” double-click the object assigned under the “Controller” variable (this is the corresponding “AnimatorController” object called “StarterAssetsThirdPerson”). This should open the players “AnimatorController” in the “Animator” window. Alternatively, you can search for “StarterAssetsThirdPerson” in the “Project” tab and double-click the found “AnimatorController”.

In the “Animator” window press the settings icon next to “Base Layer” under the “Layers” tab and activate the “IK pass” for the “Base Layer”.

INSTALLATION
BUILD-IN DEMO

To get started with the Build-In demo scene you have first to import the corresponding package.

From the folder "Assets_HoaxGames\iStep\" import the package:

"iStep_Demo_BuildIn.unitypackage"

REQUIRED PACKAGES

The demo scene requires the following Unity packages to be installed:

- *Cinemachine*
- *Input System*
- *Post Processing*

Typically, the within the package included "Starter Assets" subset will install those dependencies automatically through the included Editor scripts. Should there be any issues please try to resolve them by going to "Tools" → "Starter Assets" → "Reinstall Dependencies".

Alternatively, you can install the required packages (dependencies) manually using the "Package Manager". To do so go to "Window" → "Package Manager" and change the packages to show settings to "Packages: Unity Registry" in the top left corner of the "Package Manager" window (the setting is a drop-down menu).

Select "Cinemachine" from the list and press "install" (bottom right corner).

Select "Input System" from the list and press "install" (bottom right corner).

Select "Post Processing" from the list and press "install" (bottom right corner).

If you went for the manual installation, please make sure that under "Edit" → "Project Settings" → "Player" → "Other Settings" you have "STARTER_ASSETS_PACKAGES_CHECKED" added to the "Scripting Define Symbols".

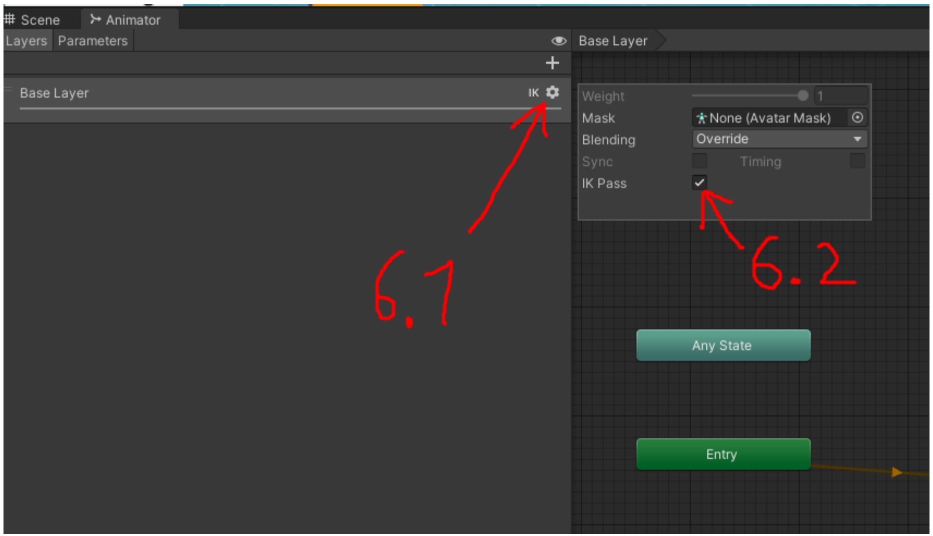
ANIMATOR SETTINGS

iStep requires an IK pass to be enabled in the Animator. To do so open the imported demo scene ("iStepShowcaseDemoBuildIn.unity") and select the gameobject "PlayerArmature" from the hierarchy. From the "Animator" component attached to "PlayerArmature" double-click the object assigned under the "Controller" variable (this is the corresponding "AnimatorController" object called "StarterAssetsThirdPerson"). This should open the players "AnimatorController" in the "Animator" window. Alternatively, you can search for "StarterAssetsThirdPerson" in the "Project" tab and double-click the found "AnimatorController".

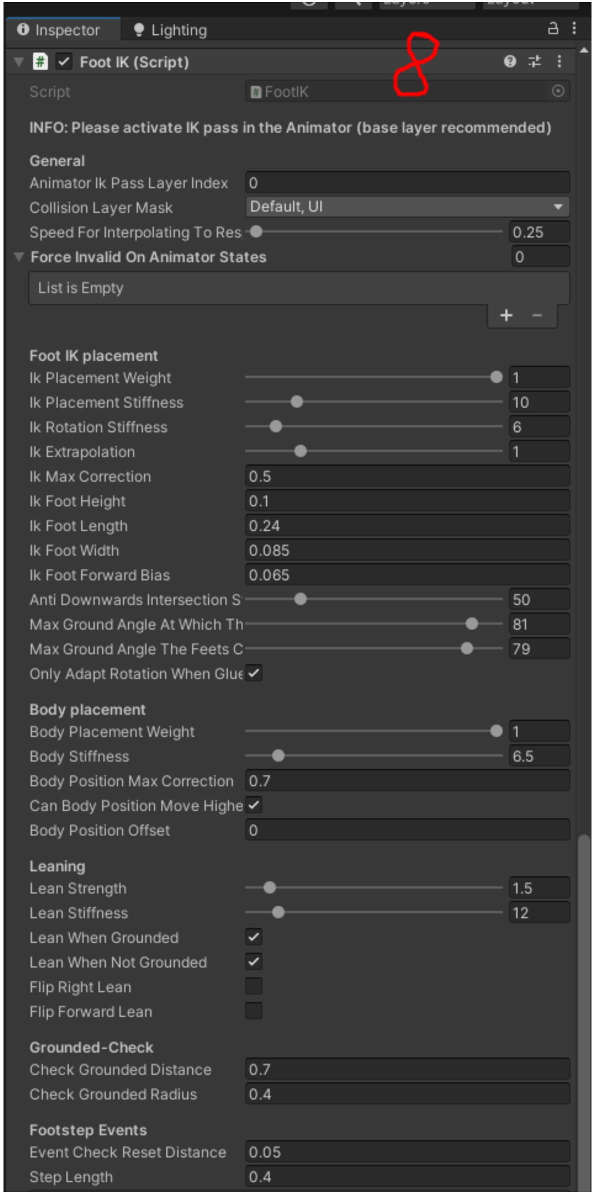
In the "Animator" window press the settings icon next to "Base Layer" under the "Layers" tab and activate the "IK pass" for the "Base Layer".

CUSTOM CHARACTER CONTROLLER INTEGRATION

Step 1: Open the corresponding Project.
Step 2: Select the gameobject representing your character controller (can be a player or an AI controlled character controller).
Step 3: Select the (child) gameobject of your character controller object-hierarchy having the Animator component attached to it.
Step 4: Add the component “FootIK” to it.
Step 5: Open the assign AnimatorController in the “Controller” variable of the Animator component by double-clicking it (the AnimatorController should open up in the Animator window).
Step 6: Make sure under the Base Layer Settings, IK pass is enabled.
Step 7: You are done. Press play to check if everything works as expected.
Step 8: Configure your “FootIK” settings to your likings if necessary.



Weight 1
Mask None (Avatar Mask)
Blending Override
Sync □ Timing □
IK Pass ☑



☑ Foot IK (Script)

Script FootIK

INFO: Please activate IK pass in the Animator (base layer recommended)

General

Animator Ik Pass Layer Index 0
Collision Layer Mask Default, UI
Speed For Interpolating To Res 0.25
Force Invalid On Animator States 0
List is Empty

Foot IK placement

Ik Placement Weight 1
Ik Placement Stiffness 10
Ik Rotation Stiffness 6
Ik Extrapolation 1
Ik Max Correction 0.5
Ik Foot Height 0.1
Ik Foot Length 0.24
Ik Foot Width 0.085
Ik Foot Forward Bias 0.065
Anti Downwards Intersection S 50
Max Ground Angle At Which Th 81
Max Ground Angle The Feets C 79
Only Adapt Rotation When Glue ☑

Body placement

Body Placement Weight 1
Body Stiffness 6.5
Body Position Max Correction 0.7
Can Body Position Move Highe ☑
Body Position Offset 0

Leaning

Lean Strength 1.5
Lean Stiffness 12
Lean When Grounded ☑
Lean When Not Grounded ☑
Flip Right Lean □
Flip Forward Lean □

Grounded-Check

Check Grounded Distance 0.7
Check Grounded Radius 0.4

Footstep Events

Event Check Reset Distance 0.05
Step Length 0.4

9

SOLVING FOR CROUCHING

In case your custom character controller supports crouching, you have two options to make iStep switch to a crouching specific behaviour back and forth.

The first and most simple option is to add your crouch-specific animation state name to the “Force Activate Crouching Behaviour” list in the inspector of the FootIK script.

The second option is to use the function “setActivateCrouchingBehaviour” from code to trigger the crouch-specific behaviour.

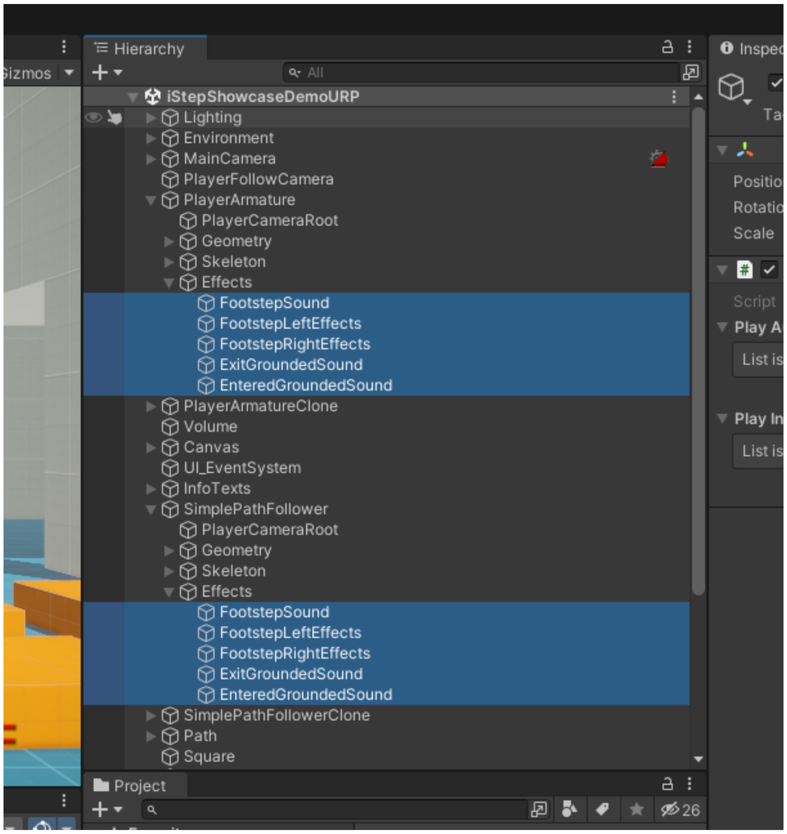
SOLVING FOR SPECIAL CHARACTER CONTROLLER ACTIONS

In case your custom character controller has any actions where iSteps foot placement solver would interfere, you can smoothly disable iStep with the following options (smoothly in this regard means that a smooth transition is applied):

Option 1: Add your action specific animation state name to the “Force Invalid On Animator State” list in the inspector of FootIK.

Option 2: Do it from code by calling “setIsValidAndShouldCheckForGrounded” function with the respective parameter.

FURTHER INFORMATION



- iStepShowcaseDemoURP
  - Lighting
  - Environment
  - MainCamera
  - PlayerFollowCamera
  - PlayerArmature
    - PlayerCameraRoot
    - Geometry
    - Skeleton
    - Effects
      - FootstepSound
      - FootstepLeftEffects
      - FootstepRightEffects
      - ExitGroundedSound
      - EnteredGroundedSound
  - PlayerArmatureClone
  - Volume
  - Canvas
  - UI_EventSystem
  - InfoTexts
  - SimplePathFollower
    - PlayerCameraRoot
    - Geometry
    - Skeleton
    - Effects
      - FootstepSound
      - FootstepLeftEffects
      - FootstepRightEffects
      - ExitGroundedSound
      - EnteredGroundedSound
  - SimplePathFollowerClone
  - Path
  - Square

All serializable parameters in the FootIK script are documented with tooltips. Please read the tooltips for more information on the respective parameters. Also, when optimizing the FootIK parameters, it is strongly recommended that you start with the default settings first and optimize from there. The FootIK script has an open interface to its footstep events. Please check out the included demo scene(s) to understand how the events were used in connection with the EffectsPlayer script to play audioclips as well as instantiate footstep effects. In the demo scene(s) all footstep effects are defined in the gameobjects selected in the above reference image.

Please note that the Effects Player script is just a way to use iSteps events but it is not limited by it.

MOST IMPORTANT SETTINGS FOR CUSTOMIZATION

- **Ik Max Correction** (Sets how much you can step up visually.)
- **Ik Foot Height** (Sets the height of your foot.)
- **Body Position Max Correction** (Sets how much you can step down visually.)
- **Check Grounded Distance** (This parameter is used by iStep to determine if the IK placement algorithm should check for new foot IK positions/rotations or not. Only when grounded is true the foot IKs will adapt accordingly. This value should typically be set as high as “Body Position Max Correction” or higher. Don’t misunderstand this parameter with your character controllers internal grounded state.)

VIDEO TUTORIALS

Getting Started with HDRP:
https://hoax-games.web.app/istep/#gettingstartedwithhdrp

Getting Started with URP:
https://hoax-games.web.app/istep/#gettingstartedwithurp

Getting Started with Build-In:
https://hoax-games.web.app/istep/#gettingstartedwithbuildin

Custom Character Controller Integration:
https://hoax-games.web.app/istep/#customcharactercontroller

iStep Settings Explanation:

https://youtu.be/pS8lrgG9C44

CONTACT

E-mail: maishofen2@hotmail.com

Website: https://hoax-games.web.app