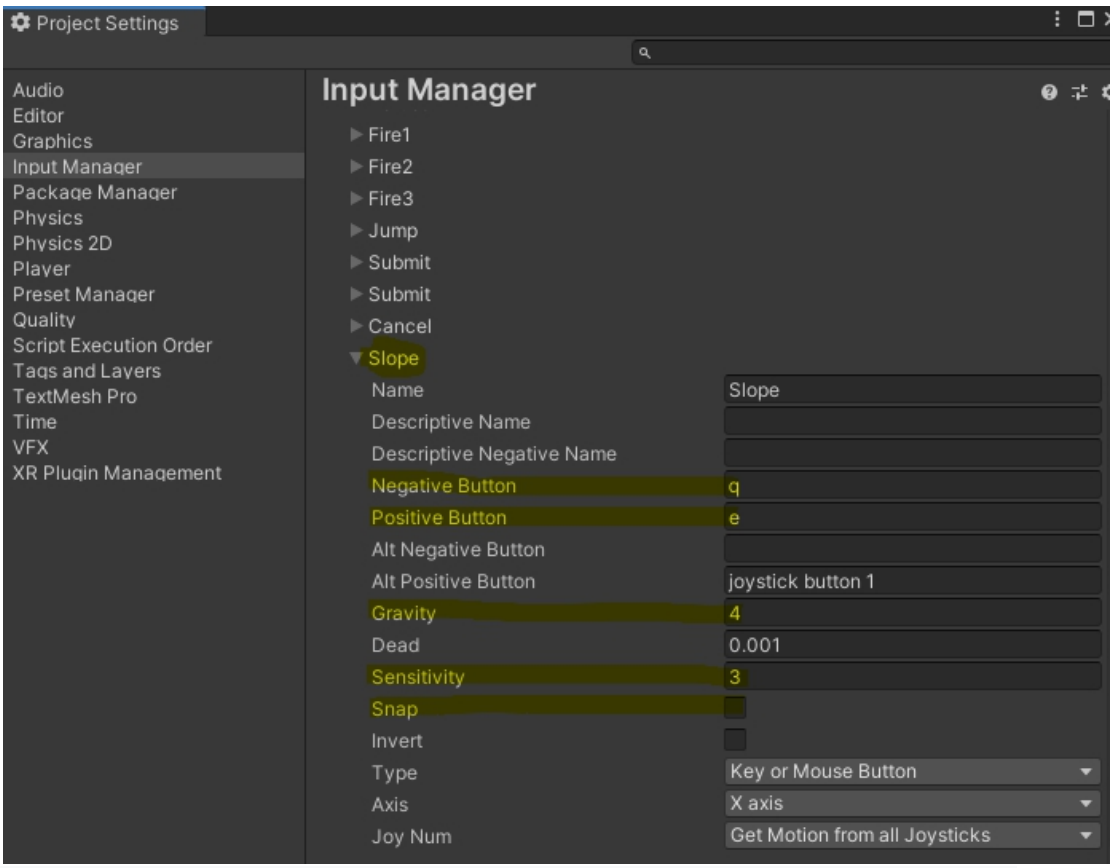
FPS TPS 2.0

1. To start
    - 1.1 RigBase.cs
    - 1.2 RigGroup.cs
    - 1.3 OnRig.cs
    - 1.4 Aim.cs
    - 1.5 BodySlope.cs
    - 1.6 PositionConstrained.cs
    - 1.7 RotationConstrained.cs
    - 1.8 RotationConstrainedForBone.cs
    - 1.9 SlotController.cs
    - 1.10 TBIK.cs

1. To start

To get started, you need to do the following after importing the package:

1. Add button "Slope" in Input manager:



2. It is desirable to disable "snap" for Horizontal and Vertical in input manager.
3. Move "Player" prefab from "Natural fps tps teamplate / Prefabs" on scene.

2.1 RigBase.cs

RigObjects	collection of objects with "RigGroup.cs"
OnRigs	all components of rigGroup objects

OnRigsInitialized() - get all the components of the rig Group objects

2.2 RigGroup.cs

jobsObjects	objects with "OnRig" component

GetOnRigsList - returns an array with all components from jobsObjects.

2.3 OnRig.cs

Execute() - all classes inherited from "OnRig" must implement "Eexecute()" for execution in "RigBase"

2.4 Aim.cs

weight	aiming weight (0 - do not aim, 1 - 100% aim)
target	Aiming target
body	Character model
rotationOffset	Rotation offset

2.5 BodySlope.cs

canSlope	enable or disable the ability to tilt the body
constrainedObject	bone for tilt
bodyTransform	Character model transform
weight	Slope weight (0 - do not slope, 1 - 100% slope)
slopeAngle	tilt angle, set a value to apply tilt

2.6 PositionConstrained.cs

constrainedType	WorldToWorld applying a global position LocalToLocal applying local position
active	enable and disable
weight	weight of position application (0 - original position, 1 - converted position)
constrained	constrained object
source	Source object
axesActive	applying position along the axes (0 - do not apply, 1 - apply)
offset	offset

2.7 RotationConstrained.cs

rotationType	type of conversion from local or global to local or global
constrained	Constrained object
source	Source object
rotationActive	applying rotation along the axes (0 - do not apply, 1 - apply)
actualActive	apply original rotation along the axes (1 - apply, 0 - return rotation to zero)

2.8 RotationConstrainedForBone.cs

source	Source object
joint	bone
body	Character model transform
weight	weight

2.9 SlotController.cs

constrained	**Constrained object**
weight	**weight**
inactiveSlot	**Slot point**
handActive	**0 - rightHand, 1 - left hand**

ApplyHandOffset(int handID, bool applyOffset) - applying the offset rifle in hand

2.10 TBIK.cs (two bone IK)

endJoint	Bone for IK
target	Target for IK
posOffset	Position offset
weight	Weight, rotation weight, position weight