Upgrading from 3.0 to 3.1

- BaseCharacterController and BaseMoverController are now interfaces (ICharacterController and IMoverController). This means you have to convert your controllers to this new approach. This can be accomplished in a few simple steps:
    - Make your controller classes inherit from MonoBehaviour (or whatever else you choose) instead of BaseCharacterController/BaseMoverController, and make it implement the "ICharacterController"/"IMoverController" interface
    - Remove the "override" keyword from all of the methods that were inherited from BaseCharacterController/BaseMoverController
    - Add a "public KinematicCharacterMotor Motor" field to your class, so you can assign the Motor in the inspector
    - In Start(), assign this class to the "Motor.CharacterController"
- PreventSnappingOnLedges was removed. Replace this behaviour by adjusting the "MaxVelocityForLedgeSnap" value (0f means it will never snap)
- ExampleCameraController rotations were fixed to be framerate-independant, so you will need to re-tweak your rotation speed values if applicable

Upgrading from 2.2.0 to 3.0

- You should manually remove the Rigidbody component on your character GameObject since it is not needed anymore (although leaving it there should not cause any problems)
- KinematicCharacterSystem.InterpolationMethod enum was changed to a simple boolean (KinematicCharacterSystem.Interpolate)
- Several field names in KinematicCharacterMotor were refactored, but they are mostly fields destined for private use

Upgrading from 2.2.0 to 2.2.1

- "SetCollisionSolvingActivation" was renamed to "SetMovementCollisionSolvingActivation"

Upgrading from 2.0 to 2.1

A complete package overwrite is required

KinematicCharacterMotor changes

- “SetStabilitySolvingActivation” renamed to “SetGroundSolvingActivation”
- “StableInteractiveRigidbodyVelocity” renamed to “AttachedRigidbodyVelocity”
- “DynamicPushForce” renamed to “SimulatedMass”
- “PlanarConstraint” renamed to “PlanarConstraintAxis”
- Added a “PostGroundingUpdate” call to the BaseCharacterController
- Ledge handling has been moved from ExampleCharacterController to KinematicCharacterMotor again
- Step handling methods are now:
    - None
    - Standard (unlimited max step height)
    - Extra (like standard, but does additional raycasts to allow better stepping on steps smaller than the capsule’s radius)

Example content changes

- ExampleCharacterController.SetInputs() now takes an inputs struct as parameter
- All example characters including Walkthrough now have a new state handling method
- ExamplePlayer now takes ExampleCharacterCamera instead of OrbitCamera

Upgrading from 1.2.2 to 2.0

A complete package overwrite is required

BaseCharacterController changes

Renamings

- The “KinematicCharacterMotor” parameter has been renamed to simply “Motor”

Changes

- ProcessHitStabilityReport() has been added as a mandatory method to implement. It is used to give you an opportunity to modify whether or not a hit can be considered “stable”
- OnGroundHit() and OnMovementHit() take a “ref HitStabilityReport” as their last parameter, instead of a “bool”
- CanBeStableOnCollider() has been removed since ProcessHitStabilityReport() can fill this role.
- MustUpdateGrounding() has been removed. KinematicCharacterMotor.SetStabilitySolvingActivation() replaces it

KinematicCharacterMotor changes

Renamings

- CharacterHeight/Radius were renamed to CapsuleHeight/Radius
- CharacterTransform was renamed to Transform
- CharacterRigidbody was renamed to Rigidbody
- CharacterCapsule was renamed to Capsule

Changes

- All grounding information is now contained in the “GroundingStatus” struct, and all previous grounding information is now contained in the “LastGroundingStatus” struct
- SetCapsuleDimensionsAuto() has been removed. You now always have to specify the capsule’s center. If you want to do an equivalent of the old SetCapsuleDimensionsAuto(), just assign (capsuleHeight * 0.5f) to the center of the capsule
- Ledge handling logic (MaxStableDistanceFromLedge, etc.…) has been moved to BaseCharacterController implementations to allow for more versatility. See the new implementation in ExampleCharacterController

- HandlePhysics() has been removed and has been separated in several methods for more clarity:
    - SetCapsuleCollisionsActivation()
    - SetCollisionSolvingActivation()
    - SetStabilitySolvingActivation()

OrbitCamera changes

Changes

- You must now call OrbitCamera.UpdateWithInput() instead of calling OrbitCamera.SetInputs(). This change was made to ensure the camera will update instantly instead on relying on component execution orders, which can vary