Release Notes

Havok Physics

Version : 6.6.0 Release

including all development since version 6.5.0 Release

Character Controller

Add asserts

HVK-5038	Implemented	**Add an assert to moveToLinearCastHitPosition to check if displacementLength is 0 which causes a Divide by Zero error** Assert added.	6.6.0 Beta

Character Controller RB

Bugs

HVK-4835	Fixed	**hkpCharacterRigidBody does not stay on dynamic box on top of platform** A demo has been created to show how the Rigid Body Character can ride on top of dynamic bodies which are on moving platforms.	6.6.0 Beta
HVK-4937	Fixed	**Overlapping character rigid body controllers are sucked together, not pushed apart.**	6.6.0 Beta
Behavior Change		The character rigid body's unwelding code now correctly takes contact normal direction into account.	

Improvements

HVK-4624	Implemented	**Rigid Body character controller should use a custom collision tolerance** The Rigid Body Character now has a softer notion of support which removes the need for adjusting collision tolerances.	6.6.0 Beta

HVK-4769	Implemented	**hkpCharacterRigidBody::checkSupport does not report correct supported state**	6.6.0 Beta
Behavior Change		The Rigid Body Character notion of support has been made "softer" so the character is less likely to lose support. Consequently, oscillations between character states have been much reduced.	

Constraints

Bugs

HVK-4979	Fixed	**Max impulse check can fail for sliding contacts**	6.6.0 Beta
		The hkpBreakOffPartsUtil occasionally failed to fire the breakOffSubPart() callback when the impulse was enough to cause a break. This failure happened rarely, and only for glancing TOI contacts. This is fixed now.	

HVK-5055	Fixed	**hkpCollisionMassChangerUtil can change the inertia of the wrong body.**	6.6.0 Beta
		A bug in hkpMassChangerModifierConstraintAtom caused this issue. Users of hkpCollisionMassChangerUtil or direct users of hkpResponseModifier::setInvMassScalingForContact could hit this bug. This bug only affected changing the apparent inertia at a collision (a new feature added in 6.5); mass changes were still correct.	

Improvements

HVK-4967	Implemented	**hkpWorldConstraintUtil::addModifier() should check for existing modifier**	6.6.0 Beta
		Assert added	

Demos

Bugs

HVK-4993	Fixed	**SlidingWorldDemo duplicates code in SlidingWorldDemo::recenterBroadPhaseVariant()**	6.6.0 Beta
		Removed redundant code in the Sliding World Demo which recalculated entity aabbs after a broad phase shift. hkpWorld::shiftBroadPhase does this already.	

hkcollide

Bugs

HVK-4936	Fixed	**hkpSpuJobDispatcher does not register hkpExtendedMeshShape's shape functions** Keyframed objects with hkpExtendedMeshShape wrapped with a MOPP caused crashes when processed on SPU. This was caused by relevant aabb-calculation functions being not registered properly.	6.6.0 Beta
HVK-4997	Fixed	**hkpTransformShape wrapped in hkpListShapes do not respect transforms** When shapes were wrapped inside a hkpTransformShape and put into a hkpListShape, child shapes that were not the first shape did not collide properly. This only happened during continuous collision detection. If DEBRIS quality was chosen, or the the hkpConvexTransformShape was used in instead, the problem did not occu. This issue is now fixed.	6.6.0 Release
HVK-5067	Fixed	**hkpWorld::addEntityBatch() can break the broadphase consistency.** Fixed.	6.6.0 Beta
HVK-4748	Fixed	**linear casting using a collision filter on SPUs** When linear casting on SPU, collision filtering callbacks could have invalid hkpCdBody hierarchies. This was caused by invalid registration of collision functions. This is fixed now.	6.6.0 Beta
HVK-4763	Fixed	**Linear casting follows a different code path on the SPUs for listShape vs. MOPP potentially causing linear casts to not pick up any contact points** Collision functions were registered wrongly on SPU. As the result, linear casting malfunctioned for hkpListShapes and failed to report contact points. This is fixed now.	6.6.0 Beta
HVK-5001	Fixed	**Linear casting hkpListShape on SPU only returns a single hit** Collision functions were registered wrongly on SPU. As the result, linear casting malfunctioned for hkpListShapes and failed to report contact points. This is fixed now.	6.6.0 Beta

HVK-4216 Fixed — **Raycasting against hkpConvexVerticesShape does not take convex radius into account** — 6.6.0 Beta

This has been fixed by introducing a new constructor for hkpConvexVerticesShape, which uses the shape shrinker to automatically shrink the vertices.
Also hkpConvexVerticesShape::setRadius() now fires a warning and should not be used, but the new constructor instead.

HVK-4775 Fixed — **hkpMoppAabbCastVirtualMachine::addHit does not do collision filtering on spu** — 6.6.0 Release

Behavior Change

Previously, hkpMoppAabbCastVirtualMachine::addHit() did not perform any collision filtering on the SPU. All collisions were reported.

A call to hkpSpuCollisionFilterUtil::s_shapeContainerIsCollisionEnabled() has been added, so collision filtering will be conducted through this function.

HVK-4867 Fixed — **Usage of hkpGroupFilter is hard coded when world linear casting on SPU** — 6.6.0 Beta

hkpSpuCollisionQueryFilterUtil's callback function are now always used for filtering of asynchronous queries on SPU: rayCasts, linearCasts, and getClosestPoints calls.

HVK-4876 Fixed — **hkpMoppCachedShapeMediator::setSplittingPlane Directions doesn't accept cylinder or capsule shapes in its convex check** — 6.6.0 Beta

hkpMoppCachedShapeMediator::setSplittingPlaneDirections() now accepts all convex shapes.

hkdynamics

Bugs

HVK-5022 Fixed — **hkpWorld::updateCollisionFilterOnEntity() can crash.** — 6.6.0 Beta

hkpWorld::updateCollisionFilterOnEntity() could crash for hkpEntities with multiple collision entries. This is fixed now.

HVK-4983 Fixed — **The restitution code can introduce energy** — 6.6.0 Beta

Behavior Change

The restitution code contained a bug which could cause energy gain which was particurlarly noticable for long thin objects. This code was only run on non-spherical objects where restitution was higher than .3, and where collision velocities were higher than hkpWorldCinfo::m_contactRestingVelocity. This issue is now fixed, and objects do not exhibit energy gain from high restitution values.

HVK-4999 Fixed — **The cached bounding volume data is not initialized for static objects** — 6.6.0 Beta

Fixed. Now for fixed bodies the cached bounding volume will be calculated when the object is added to the world

Improvements

HVK-4310 Implemented — **hkpEntity is now larger than 512 bytes** — 6.6.0 Beta

We have tuned the pool memory to allow for 544 bytes allocations.

Multithreading

Bugs

HVK-3036 Fixed — **Broadphase border executes a critical operation resulting in non-deterministic simulation** — 6.6.0 Release

hkpBroadPhaseBorder could cause nondeterminims as it freezes or removes bodies in non-deterministic order. You can now choose to postpone those operations until the end of the simulation step. Operations are then sorted and executed in a deterministic order. This is done by setting the hkpWorldCinfo::m_mtPostponeAndSortBroadPhaseBorderCallbacks flag.

HVK-4969 Fixed — **hkpPairLinearCastJob doesn't increment the m_jobDoneFlag** — 6.6.0 Beta

hkpPairLinearCastJob now calls hkpCollisionQueryJob::atomicIncrementAndReleaseSemaphore when it is finished, instead of just releasing the semaphore.

PLAYSTATION(R)3 Optimization

Bugs

HVK-5085	Fixed	**hkpSpuCollideUtil::initializeCollisionDispatcher could register the incorrect agents on SPU**	6.6.0 Release
		A bug which could cause the incorrent agents getting registered for a pair of shape types was fixed.	

Preprocess utilities

Bugs

HVK-2438	Fixed	**hkGeometryUtility::createConvexGeometry() produces incorrect hulls (drops significant features)**	6.6.0 Beta
		Fixed with new convex hull implementation.	

Tool Chain

Bugs

HVK-4685	Fixed	**CreateRigidBodies filter crashes on a dynamic mesh with zero radius**	6.6.0 Beta
		This has been fixed. For details see HVK-3633.	

Utilities

Bugs

HVK-4781	Fixed	**Unnecessary addReference() in hkpBreakOffPartsUtil() causes memory leaks**	6.6.0 Beta
		Not a bug. There is a removeReference in the hkpBreakOffPartsUtil() when the world gets deleted. The reference manual has been updated.	
HVK-3633	Fixed	**hkInertiaTensorComputer::computeShapeVolume MassProperties doesn't handle triangles with zero radius correctly**	6.6.0 Beta
		We have fixed a bug where a convex radius of 0.0 could cause the engine to miscalculate the volumes for a mesh's triangles. This would lead to a miscalculation of the shape's mass and result in an invalid inertia tensor with all values set to zero (Assert 0x11a9ad41).	

HVK-4988	Fixed	**Infinite loop in hkpBreakOffPartsUtil::LimitContactImpulseUtil::findWeakestPoint**	6.6.0 Beta

hkpBreakOffPartsUtil::LimitContactImpulseUtil::findWeakestPoint would loop indefinitely if different max impulses were specified for different shape keys. This has been fixed.

VDB Viewers

Bugs

HVK-5078	Fixed	**HK_PROPERTY_DISPLAY_SHAPE as NULL only used for local display, VDB ignores the property it if it is null**	6.6.0 Release

Behavior Change

The shape display viewer for the VDB and the demos now respect the following rigid body properties:

HK_PROPERTY_DISPLAY_PTR (0x1234) : a hkpShape* used for local (demos etc) display only (ignored by VDB)

HK_PROPERTY_VDB_DISPLAY_PTR (0x1235) : a hkpShape* used for VDB and local if local not set already above.

HK_PROPERTY_DISPLAY_SHAPE (0x1135) : a hkpShape*, used for both local and VDB if above not set, and removeReference is called on the shape once it is used (and the prop removed).

Improvements

HVK-5013	Implemented	**hkpPhysicsContext should register hkpWeldingViewer**	6.6.0 Beta

The Welding Viewer was not registered by the physics context, and therefore was not available by default in the Visual Debugger. This has been fixed.

Vehicle

Bugs

HVK-4843	Fixed	**Vehicle SDK wheel contact point has garbage W (seperating disatance) value**	6.6.0 Beta

hkpVehicleInstance::m_wheelsInfo[X].m_contactPoint.getDistance() now longer is unitialized. It now gets set to the suspension length;

New Features

HVK-4925	Implemented	**Support for multithreaded wheel raycasts (and linear casts) for vehicles** Support is now provided for multithreading the wheel collision detection for a collection of vehicles.	6.6.0 Beta

Interface Change