

We are very pleased to bring you the 5.5.0 Preview 3 release. Our Dev Team have been busy improving on the earlier previews that were released on 20 March 2018 and the 03 May 2018.

As always we really do welcome further feedback and comments from the CRYENGINE Community, but please can these come through our normal channels i.e. the [Github Issue Reporter](#) and the [5.5 Feedback Thread](#).

## Accessing the 5.5.0 Preview 3 Release

Engine versions are now available through the CRYENGINE Launcher.

1. Log into the CRYENGINE Launcher
2. Go to **LIBRARY -> My Engines** where your currently installed and available Engines are listed. From there Engines can be updated (installed).

## Github

Preview Engine versions are also available from Github, however Sandbox Editor source code is only available via Github.

1. Go to: [https://github.com/CRYTEK/CRYENGINE/releases/5.5.0\\_preview3](https://github.com/CRYTEK/CRYENGINE/releases/5.5.0_preview3)
2. **Download CRYENGINE\_preview\_5.5.0.193\_pc.zip**
3. **Unzip it somewhere and open the directory "CRYENGINE\_preview\_5.5.0.193\_pc"**
4. Double click on **InstallEngine.bat**

For more information. see the [Important CRYENGINE 5.5 Data and Code Changes](#).

If you are updating from CRYENGINE 5.4, please read this topic: [Migrating from CRYENGINE 5.4 to CRYENGINE 5.5](#).

- **UI:** Dependency Graph window is not responsive when opening from a selection window. **WORKAROUND:** It can be closed from the Windows tray bar, however if you don't have separate icons enabled for the different editor windows on the tray, then this can be an issue. **NOTE:** Opening the Dependency Graph say for example from the Asset Browser or from the global menu **Tools -> Dependency Graph** works fine.
- **RENDER: (PARTICLE)** Particle 'new\_year' is not being rendered correctly.
- **RENDER: (EDITOR ONLY)** Shadow square following camera.
- **RENDER:** Some brushes could flicker directly after level load.
- **RENDER:** SkyBox could be broken after several level loads.
- **CRASH: (VS2017)** After compiling and running the Engine with VS version 15.7 the Engine will crash (exception thrown), the problem originates from the last command in ( Editor\Python\plugins\crytools\startup.py ). For more information about this topic click [here](#).
- **C#:** Using the CRYENGINE C# extension for Visual Studio 2017 to start the GameLauncher, Sandbox Editor or Server can cause an error when used in the latest version of Visual Studio 2017.

## Animation General

- **Fixed:** Attachments (e.g. pistol, shotgun) wrongly attached if compute skinning is used.
- **Fixed:** Animation update not reusing its memory pools when no level is loaded (resulting in perpetual growth of memory usage when working with external tools such as Character Tool, Mannequin etc).
- **Fixed:** Race-condition crash in PoseAlignerChain. This crash occurs occasionally when using ground alignment IK.
- **Fixed:** Audio-event (which was added to an animation in the Character Tool) is not played in Editor/Game Mode.
- **Tweaked:** Introduce API example for ISkeletonAnim::StartAnimation.
- **Tweaked:** Added Mannequin API examples.
- **Tweaked:** Added API examples for pose modifiers and motion parameter.

## AI System

- **New:** Islands connectivity is taking navigation annotations into account.
- **New:** Debug draw functions are getting navmesh query filter from NavigationComponent with name "MNMDebugQueryFilter".
- **Refactored:** (GameTemplates) Pathfinder Component replaced by Navigation Component.
- **Refactored:** Added base interface classes to AI Components so they could be created and used in code.
- **Refactored:** Added NavMesh query filter to all functions from Navigation system that need it.
- **Fixed:** Registering smart object as off-mesh link doesn't work after creating a new level.
- **Fixed:** (Components) Completing and fixing Navigation Markup Shapes Component, synchronizing with exported data, more shapes possible per Component.
- **Fixed:** Crash when generating navmesh on dense terrain.
- **Fixed:** Crash in canceling AI sequence when actor was no longer valid.
- **Fixed:** Crash in Raycast when visiting degenerate triangle.
- **Fixed:** Compilation error (warning treated as an error).
- **Tweaked:** Support for older navmesh versions being loaded.

## Middleware Updates

- Updated to Wwise SDK v2017.2.4
- Updated to Fmod Studio 1.10.05
- Updated Oculus spatializer for Wwise from 1.17.0 to 1.20.0

## Audio General

- **New:** (SDL Mixer) File import for external audio files into ACE.
- **New:** (SDL Mixer) Audio parameter and switch implementation for SDL Mixer.
- **Optimized:** Smoother scrolling and faster expanding of treeview items in ACE.
- **Fixed:** A non-thread-safe member variable of audio objects was accessed in a multithreaded fashion, this is no longer the case and the member variable is now only accessed by the main audio thread.
- **Fixed:** Wwise implementation now uses proper MSVC libraries during linking in regards to the used compiler toolset.
- **Fixed:** Wwise implementation did not load the convolution reverb plugin correctly.
- **Fixed:** No account taken if occlusion results should be accumulated or not during very first sample run.
- **Fixed:** Audio listeners not updating their name if the owning entity's name was changed.
- **Fixed:** Parameters were not updated in Fmod 1.10.02
- **Fixed:** False error warnings (because of empty connections of "do\_nothing" trigger). Added class CDoNothingTrigger and execute triggers on CATLTriggerImpl instead of CATLAudioObject to prevent error.
- **Fixed:** Added missing warnings for duplicated switches and parameters.
- **Fixed:** Added missing warnings about unknown XML tags to middleware implementations.
- **Tweaked:**New: Occlusion rays now use the new "rwi\_max\_piercing" flag to be independent from a surface's "piercability" attribute, additionally rays can now continue until the 10th hit (formerly 5).
- **Tweaked:** Audio Listener Component received the parent entity's name (which confused people). Now named to just "Audio Listener" however, the underlying listener object still has the following naming format "audio\_listener\_<entityname>\_<entityId>" for debugging reasons.

## DRS (Dynamic Response System)

- **Fixed:** Dialog Line Database entries are overwritten as soon as any new text is entered.

## Engine General

- **Refactored:** Removed small primitive shapes.
- **Refactored:** Remove now unsupported GameZero cryproject.
- **Optimized:** Added early out for IEntitySystem::InitEntity should init fail and you want to delete the entity (avoids going through all deleted entities).
- **Fixed:** It is not possible to queue fragments from C++ in the Advanced Animation Component.
- **Fixed:** It is possible for users to add input devices but not remove them (resulting in shutdown crashes).
- **Fixed:** Entity Component replication order having changed - resulting in GameSDK multiplayer sessions not starting correctly.
- **Fixed:** Where Entities marked as client only, server only or not replicated could still be replicated over the network.
- **Fixed:** Entity simulation mode being started during Level Load in Sandbox Editor.
- **Fixed:** Loading any map in pure game results in crash CEntityLoadManager::ExtractCommonEntityLoadParams.
- **Fixed:** Invalid shutdown order unit test.
- **Fixed:** Invalid assert that could be fired if an entity requests updates during its own update loop.
- **Fixed:** Where OnComponentMaskChanged could be called when mask did not change.
- **Fixed:** Where an entity could get resurrected when entering/exiting Game Mode, but would still be stored in the deleted entities vector and get removed next frame.
- **Fixed:** Where Components would not be shut down.
- **Fixed:** Possible double-deletion if game code attempts to remove an entity from its ENTITY\_EVENT\_DONE event.
- **Fixed:** IEntity::GetSerializableNetworkSpawnInfo's callback was not thread-safe.
- **Fixed:** Entity event masks were not always correctly updated - caused a crash when spamming save/load in GameSDK.
- **Fixed:** Possible crash if a Game Object is removed during post update.
- **Fixed:** A client is not able to join a Rolling Ball Multiplayer Game.
- **Fixed:** C++/top down shooter camera rotates at high speed if the project is started in Game Launcher.
- **Fixed:** IEntityComponent::OnTransformChanged could be called before IEntityComponent::Initialize.
- **Fixed:** Entity Component GUIDs being reset every time a map is loaded in the Sandbox Editor.
- **Fixed:** IEntityComponent::NetReplicateSerialize being called directly on Component creation and before entity initialization - causing a possible crash. Now serializes in bulk once entity is initialized.
- **Fixed:** Inability to create a physics primitive and mesh colliders with 0 mass (resulting in immovable entity that still reports collisions).
- **Fixed:** Entity Audio Component not respecting its own local transformation (always playing audio at the entity position).
- **Fixed:** Audio Listener Component playing audio at the entity position and not using the Component transform.
- **Fixed:** Case where replacing an Entity Component would leave a dangling pointer in update listeners.
- **Fixed:** New Components no longer being able to receive immediate collision events.
- **Fixed:** Every logged collision event requiring a look-up of the script proxy (iterate through all Components) for both source and target Entities.
- **Fixed:** Action map groups could override the same action name.
- **Fixed:** Potential crash when sorting Entity Event Listener
- **Fixed:** Issue where Components could be in two different Editor categories with the same name.
- **Fixed:** Issue where Components could not be created using a base interface ID.
- **Fixed:** Null pointer dereference when using Chinese IME on Windows 10.
- **Tweaked:** Move EEntityEvent to namespace for Doxygen documentation, mark global namespace event as deprecated.
- **Tweaked:** Move generate Cubemap button to the top of the Environment Probe Component.
- **Tweaked:** Introduce additional API examples.
- **Tweaked:** Introduce API examples for IEntity::Load\* functions.
- **Tweaked:** Added API examples for preview rendering.
- **Tweaked:** Added 'applyImmediately' option to CAdvancedAnimationComponent::SetCharacterFile, always load new character from disk.
- **Tweaked:** Replaced the normal primitive shapes with 1 unit sized geometry.

## Common

- **New:** Introduce new CryAPIExamples module to compile Doxygen snippets.
- **New:** Introduce action map listener example.

- **New:** Add examples for common physics functions.
- **New:** Introduce networked client listener example.
- **New:** Introduce minimal plug-in example.
- **New:** Add simple IEntityAudioComponent examples.
- **New:** Add XML & Json examples.
- **New:** Add asynchronous camera injection (VR) example.
- **New:** Add Trigger and Rope Component examples.
- **New:** Add ISkeletonPose::GetAbsJointById example.
- **New:** Introduce CVar and CCommand examples.
- **New:** Add ENTITY\_EVENT\_COLLISION example.
- **Fixed:** CTransform::ToMatrix34 not respecting scale correctly.
- **Tweaked:** Document IPersistentDebug.
- **Tweaked:** Document IProjectManager.
- **Tweaked:** Move Action Map Manager interface to CryCommon.
- **Tweaked:** INetContext, IRenderer and IPhysicalWorld not having Doxygen pages generated.
- **Tweaked:** Added default values to the SFogVolumeProperties structure.
- **Tweaked:** IStatObj not using correct Doxygen style comments.
- **Tweaked:** Move Mannequin interfaces to CryCommon for Doxygen support, document common types.
- **Tweaked:** SNetObjectId not receiving Doxygen comments.
- **Tweaked:** Add examples for RMis and aspect delegation.

## System

- **Fixed:** INumberVector.NormalizeSafe: assignment to \*this requires cast to final type.
- **Fixed:** Disable plugin load from disk if shared libraries are unsupported (fixes PS4 startup issue).
- **Fixed:** Alt tab mouse cursor window lock.
- **Fixed:** Mouse cursor visible on alt tab. Fix double cursor visible on startup.
- **Fixed:** Command line not being executed when running the shader cache generator.
- **Fixed:** CVar whitelist not working.
- **Fixed:** OOM handling when cleaning up bucket allocator.
- **Fixed:** Incorrect Engine root directory being used for project manager on PS4.
- **Fixed:** Non-lowercase paths being used in project manager, failing when 'sys\_project=GameSDK' is used instead of 'sys\_project=gamesdk'.
- **Tweaked:** Added threads to thread config.
- **Tweaked:** Document ISystem::GetCamera and SetCamera.
- **Tweaked:** (Linux) Use clang-3.9 and gcc-6 by default.

## CMake

- **New:** Support building API examples without the Engine for auto compilation.
- **Fixed:** Pass the build type to RC when building as an external project.
- **Fixed:** Inability to build shader cache generator from a project solution.

## Action General

- **Fixed:** By-ref access to CVar value contained in a removed system.
- **Fixed:** Game Launcher crashes on quit after fighting AI (CryAction.dll!CAIHandler::QueryComplete).
- **Fixed:** Jumping into Game Mode on certain maps leads to crash (CGameObjectSystem::PostUpdate()).
- **Fixed:** ENTITY\_FLAG\_LOCAL\_PLAYER being replicated over the network.
- **Fixed:** PersistentDebug objects not rendering when the timeout is set to 0.

## Schematyc

- **Fixed:** Inability to use Schematyc resource types in a Schematyc array.

## Templates

- **Fixed:** Rolling Ball template not synchronizing player's physical states over the network.

## Default Entities

- **New:** Add preview drawing to the Physic Constraint Components.
- **New:** Geometry and Physics Components with Mass and Density now have an interactive dropdown to select either Mass, Density or Immovable.
- **Fixed:** Networked property of the Rigidbody Component not having any effect.
- **Tweaked:** Add preview rendering if the Mesh Component is only used as collider.
- **Tweaked:** Changed the default FOV for the Camera Component to 70.

## Renderer General

- **New:** Profile section to detect GetOrCreateInputLayout's shader reflection request.

- **Refactored:** Use correct frameid for Oculus.
- **Refactored:** Shader modifications made to support new HLSL to SPIR-V shader compilers for Vulkan.
- **Refactored:** Renamed ColorMask::Count to GS\_NOCOLMASK\_COUNT to prevent it being a globally defined symbol with that name.
- **Refactored:** To add a new platform - now just add shaderlist file name to the GetShaderlistName function.
- **Refactored:** Replacing r\_shadersorbis, r\_shadersdx10, r\_shadersdx11, r\_shadersGL4, r\_shadersGLES3, r\_shadersdurango, r\_shadersVulkan CVars with r\_ShaderTarget CVar - which specifies the target which shaders must be built for.
- **Refactored:** (Vulkan) Trigger a fatal error when the resource layout encoding does not have enough space to store it.
- **Refactored:** (Vulkan) Proper fatal error message when two resource layout hashes collide.
- **Refactored:** Unify swapchain creation paths, now exclusively handled by render display context.
- **Fixed:** AI's skin is not rendered.
- **Fixed:** Cubemap generation: Disable LOD transition.
- **Fixed:** Display context resize.
- **Fixed:** (DX12) Device removal due to slightly mismatched texture copy targets.
- **Fixed:** Choose closest matching resolution as early as possible.
- **Fixed:** Vulkan fullscreen broken due to window re-position via Windows API with AMD drivers.
- **Fixed:** hmd\_resolution\_scale CVar triggers assert.
- **Fixed:** Activating NVIDIA multi-res triggers assert (note this feature is still WIP and as yet is not fully functional!).
- **Fixed:** Setting the global aux geometry command buffer when changing the aux geometry command buffer collector (was causing a race condition).
- **Fixed:** Crash when CV\_r\_stat == 6 but pDrawCallInfoPerNode is not yet filled.
- **Fixed:** Shader reflection memory leak (was causing crash in Vulkan after calling r\_reloadshader 1 a few times).
- **Fixed:** Drop stereo socialscreen PSOs at unload.
- **Fixed:** Sending the correct shaderlist filename to remote shader compiler for Vulkan shaders.
- **Fixed:** (DX12) Lost device state when transitioning into fullscreen from windowed and minimized state.
- **Fixed:** Sys\_job\_system\_proiler 1 aux drawings.
- **Fixed:** Crash when deleting water volumes because of immediate release of CRElement.
- **Fixed:** (Xbox One) D3D memory leak.
- **Fixed:** (PS4) "r\_driver should match shader target" (fatal error issue for PS4 shader cache generation).
- **Fixed:** (PS4) Shader cache generation failure. r\_Driver=Dx11 is allowed for Orbis shader target flag.
- **Fixed:** Vulkan shader cache generation. The encoded resource layout size is reduced by half. It is base64 encoded and sent with request line to remote shader compiler. This way the resource layout description which is needed for Vulkan shader compilation is known during shader cache generation.
- **Fixed:** Permanent render object's general pass transform to be overwritten by shadow passes.
- **Fixed:** The textured aux geometry color (was carried out once in the vertex shader and once in the pixel shader. Once in the pixel shader will just cancel out the transformation done in vertex shader).
- **Fixed:** Crash when clicking generate Cubemap in the Sandbox Editor.
- **Fixed:** Low spec crash in the Sandbox Editor.
- **Fixed:** Issues that occur when lots of aux draw commands are being executed.
- **Fixed:** Aux rendering lag and flickering issue in Character Tool.
- **Fixed:** Screen space shadows missing in the background.
- **Fixed:** Aux geometries depth rendering issue.
- **Fixed:** Cursor may get stuck in previous client window dimensions after changing the resolution.
- **Tweaked:** Remove PostAA assert, condition is handled.
- **Tweaked:** Reduce flickering when entering fullscreen (using native resolutions by changing swapchain fullscreen state switch order to MS recommended approach).
- **Tweaked:** Improve multi-monitor window management, automatically switch output when the window is moved onto a new monitor.

## Vulkan

- **New:** Annotating HLSL shader to include resource layout descriptors.
- **New:** Local shader compilation in Windows platform.
- **New:** HLSLcc local shader compilation.
- **New:** Support for DXC and GLSLANG HLSL to SPIR-V compilers and compiler switching feature.
- **Refactored:** Support for using new SPIRV-Cross.
- **Refactored:** Adjustments to support shader cache generation and remote shader compiler with the new shader compilers.
- **Refactored:** Changing the swap chain's color buffer format from RGBA to BGRA (validation errors were triggering because of RGBA formats on both NVIDIA and AMD).
- **Refactored:** SVOGI's full resolution kernels are now utilized for Vulkan.
- **Fixed:** NAN-Values appearing due to undefined behavior of pow(x, y) when x < 0.
- **Fixed:** The DXC compilation error for particles (fails since it tries to modify a global variable which is invalid).

## 3D Engine

- **Optimized:** Cached shadow maps are ported to one-pass scene graph traversal.
- **Optimized:** Per-objects shadow maps ported to one-pass scene graph traversal.
- **Fixed:** (Renderer) Reintroduced e\_screenshot and minimap feature.
- **Fixed:** Transformation/rotation issues for objects with sub-objects.
- **Fixed:** Brushes rendered in camera space do not take camera rotation into account.
- **Fixed:** Defer deletion of rendernodes by one frame to make sure they can be safely used in the renderer.
- **Fixed:** Assert for MovableBrushes ending up inside CObjManager::RenderObject.
- **Fixed:** All Entities disappearing (if too many shadow casting lights are visible).
- **Tweaked:** Added sensible default values to rope render node parameters (same as the Sandbox Editor uses).

## Particles

- **New:** Enhanced grouping capabilities in new ActivateRandom feature; now supports all possibilities of old SecondGen features.
- **New:** Replaced SecondGen features (on parents) with child features (on children, who decide when they activate). Replaced SecondGen "Random" parameters with new ComponentActivateRandom feature. Added AddSubInstances and CullSubInstances features, removed KillParticles feature. Parent relationship is now explicitly serialized. Added Component-to-Component node linking in Particle Editor.
- **New:** Pre-compute effect timings for stability, equilibrium and death.
- **New:** Added screen-fill stats.
- **New:** SpawnCount now has option for Max or Total particles emitted.
- **Refactored:** ParticleDataType is now type-templated - provides more type-safety and easier interface with Container Streams; now supports Vec3 etc. directly rather than float3. Streams now have [] operators. Vec3 and Quat streams are template specializations. Moving from specialized stream names to generic TStreams and TIOStreams. Simplified ParamMod Init and Update functions.
- **Refactored:** Accessing instance data is automatically templated. SpawnEntries arrays now temporary. Spawn features now have virtual GetSpawnCounts(), rather than giant templated SpawnParticles. Particle count scale from SpawnParams etc. now properly applied after GetSpawnCounts calculation. Added Runtime.GetEmitLocation.
- **Optimized:** Simplified particle age and state handling. Removed redundant State field; state is inferred from age. All particles now consistently live through the frame of their max age (previously only newborns did). All particles are rendered, no longer keep particles unrendered for 2 extra frames. SpawnOnDeath child generation and Audio trigger generation now occur on same frame.
- **Optimized:** Sped up job scheduling; use faster test for prioritizing visible emitters. Don't update unseen stable emitters (when e\_ParticlesThread = 4, now default). Added emitter and effect time variables to track emitter stable state affected by entity or attribute values. Non-updated emitters use maximum stable bounding box. Refactored stats to be correct even without emitter updates. Fixed some related bugs.
- **Optimized:** Implemented fill area limiting for pfx2. e\_ParticlesMaxScreenFill sets maximum number of screens drawn.
- **Optimized:** Test emitter water intersection before submitting RenderObjects, reducing their count.
- **Optimized:** Improved job scheduling and synchronization. e\_ParticlesThread = 2 lets update jobs sync later and allows ComputeVertices to perform update itself rather than waiting. e\_ParticlesThread = 3 reschedules emitter jobs by visibility and gives ComputeVertices priority.
- **Optimized:** Moved some bounds computation to main thread for consistency, simplified expansion function.
- **Fixed:** Particles with 0 lifetime not dying. Fixed div-zero in RenderRibbons axis computation. Fixed erroneous conversion of FeatureSecondGen random child selection. Fixed conversion of pfx1 Tail parameter to use new Child feature.
- **Fixed:** Serialization fixes: Consistently serialize all Param values including if default. Increased archive version number - based on new Child features. Added error messages for unsupported version numbers and nonexistent features.
- **Fixed:** Further fixes for discarding changes in Particle Editor.
- **Fixed:** Crash on adding invalid Child feature to emitter features. Refactored feature registration to remove invalid features from emitter features list.
- **Fixed:** Issues in pfx1 conversion: Fixed TValue serialization to remove improper zeroing of not-found values (caused ViewDistanceMultiplier to always be 0); updated for new Child features; added default SortMode; reduced intermediate XML attributes.
- **Fixed:** Further fixes for editing effects with live emitters.
- **Fixed:** Correctly unregister emitters when resetting them during sys\_spec change.
- **Fixed:** Crash when editing effects destroyed child hierarchy.
- **Fixed:** Crash when adding new Components to running effect.
- **Fixed:** Default Sprite Component had all lighting values = 0.
- **Fixed:** Editing effects again updates existing emitters; editor uses standard particle system to load effects instead of creating duplicate.
- **Fixed:** Particle age is now consistently end-of-frame age for both Update and Render sections and is always >= 0. Fixed occasional case of particles spawning before emitter delay elapsed.
- **Fixed:** Invalid lifetime values, invalid init values in padded particle data.
- **Fixed:** MoveRelEmitter to work with long update times, use sub-frame parent orientation.
- **Fixed:** Instances of GetRandomPoints returning bad values when meshes not present.
- **Fixed:** Ribbon ConnectToOrigin option.
- **Fixed:** SubInstance being initialized before emitter attributes and features are set. Runtime creation is deferred until next Update; SubInstance creation is now performed in Update jobs.
- **Fixed:** Editing effects now properly updates emitters.
- **Fixed:** Fixed and simplified modifier context handling. ModInherit now has a separate base type for proper class factory population.
- **Fixed:** Invalid float data in Spawn feature. Fixed zero-normalize in shape areas.
- **Fixed:** Crash caused by improper destruction order.
- **Fixed:** Restarting of inactive emitters on sys\_spec change.
- **Fixed:** Release all emitters in ClearRenderResources.
- **Fixed:** Properly re-create RenderObjects on sys-spec change.
- **Fixed:** Only create RenderObjects as needed.
- **Fixed:** Drag was ignored with some feature settings.
- **Fixed:** Potential null access in anim attachments.
- **Fixed:** Calculation of GPU max particle counts, including extra frame for particle life. Restore FeatureSpawnCount.duration default to 0.
- **Fixed:** Several errors involving Emitter Features. Altered effect is now completely cloned for proper parenting. Component.SetParent cleans up children. Clearing Emitter Features now updates emitter. Emitter Features no longer flagged as changed when empty.
- **Fixed:** Emitter Activate issues. Parent particle now added only once.
- **Fixed:** Broken Audio Component.
- **Fixed:** Crash caused by referencing deleted Components (e.g. during editing). Prevent registering empty emitters.
- **Fixed:** Zero particle submesh data when editing effect preventing dangling reference. Added SParticleDataTypeInfo.zeroClear option, simplified EParticleDataType declarations.
- **Fixed:** Potential JobManager crash when changing levels.
- **Fixed:** Accidental changes to acceleration and velocity effectors.
- **Fixed:** Clean up IPhysicalEntity and IAudioObject data when killing emitters.
- **Fixed:** Collision checks no longer move particles above terrain. Fixes SpawnOnCollide effects underground.
- **Tweaked:** More useful information in effect bounding box debug display. Documented unconverted pfx1 params.
- **Tweaked:** Removed obsolete Particles/Parent/timer.pfx; no effect template is needed to create child effects.
- **Tweaked:** e\_ParticlesDebug u+ now disables CameraDistance modifiers only for PerInstance params.

## C#.Core

- **Refactored:** Matrix3x3, Matrix3x4 and Matrix4x4 now only provide access to the values by index (instead of direct access to the values).

- **Refactored:** More values are now exposed in CollisionEvents.
- **Fixed:** The C# Editor plugin can now detect the installed version of Visual Studio 2017 and sets this as the default editor for C# files.
- **Fixed:** Mono being initialized when not used (causing possible shutdown issues).
- **Fixed:** Issue where C# assets had to be compiled twice at startup.
- **Fixed:** Compiler errors at startup not being shown correctly.
- **Fixed:** C# plugins not initializing correctly.
- **Fixed:** EntityProperties are now initialized with the value that's assigned to them (instead of the default value of the type).
- **Fixed:** Opening C# solutions for projects that contain a space in the filepath.
- **Fixed:** A race condition in the CharacterAnimator that could result in a crash when the C# assemblies were reloaded.
- **Fixed:** Update Mono from v5.2.0 to v5.10.0 to resolve memory corruption occurring after deserialization.
- **Fixed:** Case where the Engine attempted to copymscorlib to a temporary directory before loading.
- **Fixed:** Plugins being loaded with relative paths, resulting in a failure to later compare file locations correctly.
- **Fixed:** Case where the library vector could contain multiple occurrences of the same library due to mismatching assembly and image.
- **Fixed:** Case where class methods could be duplicated, resulting in unnecessary serialization.
- **Tweaked:** Exposed get and set methods for getting float and Vector3 parameters from materials.
- **Tweaked:** Vector2, Vector3, Vector4, Quaternion, Angles3 and Color are now valid EntityProperties.
- **Tweaked:** Added SurfaceId, PartId and PartIndex to the RaycastHit.
- **Tweaked:** Moved loading the Character file from the Player Component to the CharacterAnimator Component.
- **Tweaked:** Exposed more picker types (Character, ActionMapName and ActionMapActionName) for C# properties.
- **Tweaked:** Link to the Mono static library (removing the need to ship mono-2.0-sgen.dll).
- **Tweaked:** Exposed IsInSimulationMode property in the Engine.

## Physics

- **Refactored:** Added a description to some interface functions.
- **Fixed:** Collision issue with boxes perfectly aligned with terrain.
- **Fixed:** Memory leak in BakeScaleIntoGeometry.

## Network

- **Tweaked:** Document INetChannel.

## CryVersionSelector

- **New:** A backup of a project can now be created in the CryVersionSelector.
- **New:** When switching Engine version - the user now has the option to automatically create a backup of the project before switching Engine version.
- **New:** Update package build options with export path selection, configuration selection and the option to include debug symbols.
- **Fixed:** Packaging a build failing - if the name of a folder or a file contained; a [and] character, and/or specific numbers separated by a dash were used.
- **Fixed:** Generated CMakeLists.txt containing absolute paths to the project.
- **Fixed:** Project plugin binaries being statically linked with Engine in release mode (can not occur since projects are compiled independently).

## Editor General

- **New:** GeometryCache importer.
- **New:** Added a user confirmation dialog to control the process of updating levels to the new level format.
- **New:** (Asset Browser) Allow deletion of assets via Delete key.
- **Refactored:** Added VoxelizationMapBorder GI parameter into level settings UI.
- **Refactored:** Added ShadowsSoftness parameter into GI settings UI.
- **Fixed:** (Heightmap) default heightmap texture is broken.
- **Fixed:** (GameSDK) Player is stuck after switching into Game Mode while Character Tool is playing an animation.
- **Fixed:** Removed all Preview Build strings and reverted splash image to the default one.
- **Fixed:** Popup Menu spawn arguments in Add Component Window.
- **Fixed:** Gravity volume object crashes if used without a Lua script.
- **Fixed:** Entity bounds do not take local slot transform into account.
- **Fixed:** Lazy creation of rendercontext for QViewport and fix renderoutput resizing.
- **Fixed:** Don't create a 0x0 render context.
- **Fixed:** Assert and crash may trigger when trying to navigate outside Asset Browser default directories.
- **Fixed:** Saving a pfx under a different name does not work (error unable to copy asset).
- **Fixed:** Limited vegetation size in Paint Tool to 3.0f and size variation to 1.0f.
- **Fixed:** Added limits to slopes in terrain and vegetation painting (painting on a 90% slope is unsupported).
- **Fixed:** Ungrouping an empty group crashes the Sandbox Editor.
- **Fixed:** Extracting objects from prefab crashes the Sandbox Editor.
- **Fixed:** Sandbox Editor crash Track View: CTimeline::SMoveHandler::mouseMoveEvent
- **Fixed:** Duplicating an empty entity several times and then clicking the 'Add Component' button while they are all selected triggers an assert.
- **Fixed:** 'Snap to Grid' value can't be set below 1.
- **Fixed:** Editing a Game Volume inside of a prefab/group moves the Game Volume outside of the map.
- **Fixed:** Issue where painting on a terrain when having a slope min and max value was showing erroneous results. Changed vegetation painting to use same GetAccurateSlope function that terrain painting uses.
- **Fixed:** Effects converted from pfx1 is not visible in the Asset Browser.
- **Fixed:** Crash (trying to read empty texture property).
- **Fixed:** Removed unused Perception Modifier from the AI Objects menu.

- **Fixed:** (Mannequin) FragmentEditor/TransitionEditor/Previewer: "Go to end" button does not work.
- **Fixed:** Static model resource picker not using the asset system.
- **Fixed:** A wrongly named folder is created on the Hard Disk Drive if an \*.fbx file was dragged into the Asset Browser.
- **Fixed:** Using 'Save as' to save an animation in a custom folder doesn't work.
- **Fixed:** (Character Tool) Character update on Undo/Redo.
- **Fixed:** The Legacy Environment Probe stops working after generating Cubemaps 2 or 3 times in a row.
- **Fixed:** Crash in entity link row picker where it was calling repaint on a property tree that was in the process of being destroyed.
- **Fixed:** PyThreadState can be zeroed out in ShibokenWrapper => extra check to prevent Access Violation when update event happens.
- **Fixed:** If an edit tool button has created an edit tool - it will now also remove it on destruction.
- **Fixed:** Crash if Pick Tool was active when the Editor's layout changed.
- **Fixed:** Solved issue with objects within prefabs losing their selection flags - when modified it would eventually lead to them never being deselected.
- **Fixed:** Objects within prefabs that were selected and modified would not be removed from the selection.
- **Fixed:** Modifying the outputs of a substance-instance by adding a 'Texture Output' causes a crash.
- **Fixed:** (Schematyc) Random crash while working in Schematyc (EditorCommon.dll!CMergingProxyModel::Unmount(QAbstractItemModel \* submodel)).
- **Fixed:** Moving an edge on its local axis of an AS, CV or DO deletes its faces.
- **Fixed:** Typo in MNM regeneration information box (side instead of size).
- **Fixed:** After moving a vertex of an AS, CV or DO on its local axis it changes the gizmo to world axis.
- **Fixed:** Closing the DRS window when Response History is filled with data leads to a crash.
- **Fixed:** Adding any Component to an empty entity results in a crash (CCrySignal::DisconnectById).
- **Fixed:** Crash when opening certain particle effects (DebugCallStack::FatalError).
- **Fixed:** Creating a new material in the Asset Browser triggers a floating window which just contains a Button with the label GameSDK.
- **Fixed:** Area Selection Mask does not include AreaSolid.
- **Fixed:** Adding IsDestinationReachable node to Schematyc's signal graph and right clicking in Graph View causes a crash (EditorCommon.dll! CDictionaryModel::parent).
- **Fixed:** Duplicate objects - left clicking on them in the Level Explorer and moving the cursor into the main Viewport triggers an assert (SelectionGroup.cpp / L265)
- **Fixed:** Creating a new layer in a folder will create a new layer, but selects the last layer to rename.
- **Fixed:** Prefab proxies are not active on unhidden layers after export.
- **Fixed:** Entity can be linked with itself.
- **Fixed:** Editing the outputs of a deleted substance-archive causes a crash.
- **Fixed:** (LOD) cgf files can only be browsed after the \*.pak file has been unzipped.
- **Fixed:** (Asset Browser) Creating a new level just opens the level without asking to save the previously opened level.
- **Fixed:** Texture Compiler fails with the 'file not found' error if the texture path contains spaces.
- **Fixed:** Selecting a selection mode crashes the Sandbox Editor.
- **Fixed:** Notification windows are freezing and still visible after the Sandbox Editor is closed.
- **Fixed:** Resetting the layout via the console causes a crash.
- **Fixed:** (Trackview) Copying an animation in Track View via Ctrl+C causes a crash (CryMovie.dll!TAnimTrack<STimeRangeKey>::SerializeKeys).
- **Fixed:** Adding a Track View button to an Audio Toolbar and pressing it causes a crash (EditorTrackView.dll! anonymous namespace:: PyTrackViewAddLayerNode).
- **Fixed:** Moving layers is now an undoable action.
- **Fixed:** Mouse cursor appears outside of the main Viewport during Game Mode.
- **Fixed:** Adding handlers for opening documentation help in UQS, Python Script panels, Object Create Tool, Character Tool, DRS & Environment Editor.
- **Fixed:** (Viewport) Mouse becomes active outside of the Viewport when in Game Mode and when switching back and forth out of the Sandbox Editor.
- **Fixed:** Mouse cursor appears outside of main Viewport during Game Mode
- **Fixed:** Closing the Viewport while being in Game causes Engine to crash (CViewport::GetSafeHwnd()).
- **Fixed:** Sandbox Editor legacy material loading.
- **Fixed:** Wait on render thread to have the shader loaded when material.loadshader is executed.
- **Tweaked:** Preview renderer uses entity object color instead of the blinking blue default color.
- **Tweaked:** Make sure that an entity cannot be linked with itself and warn the user if they try to do so.

## Tools General

- **Fixed:** PakShaders.bat not working with new project workflow.

## Resource Compiler

- **Fixed:** Command line parser may return an extra arg with a random content.
- **New:** Added CVar to draw full extent of last cached shadow cascade.
- **New:** Expose CVar to control number of octree nodes visited during incremental shadow cache update.
- **Refactored:** Simplified duplicate shader detection and CResFile update.
- **Refactored:** Removed unused 'ref file' code from CResFile.
- **Refactored:** Removed unnecessary conditional from ObjManDrawEntity.
- **Refactored:** Added permanent render object compilation options which can be used to disable/enable compilation stages separately.
- **Refactored:** Storing instance separate dirty flags for general and shadow passes as they were interfering.
- **Refactored:** Storing compiled flags for compiled render objects and make sure that all parts are compiled. This is added to support compilation of constant buffers shared among multiple render items.
- **Refactored:** Improvement to permanent render object handling.
- **Refactored:** Double buffering of render object transformation matrix to avoid racing between render and main/job thread.

- **Refactored:** Make nearest objects temporary render objects always. To avoid this, we need to add nearest matrix to render object which adds the complexity of the code.
- **Refactored:** Bringing back permanent render object compilation once per general and shadow render views.
- **Refactored:** Removed unused 'MergeShaders' functionality.
- **Refactored:** Remove unused 'CShaderSerialize::m\_customSerialisePath'.
- **Refactored:** Remove unused 'CShaderSerialize::m\_customSerialisePath'.
- **Optimized:** Dynamic distance shadows are ported to one-pass scene graph traversal code path.
- **Optimized:** Introduced lastUsedFrameID for StagedTextures uploads to prevent stalls on the RenderThread in CWaterStage::ExecuteWaterNormalGen and CRenderer::EF\_SubmitWind for DX11.
- **Fixed:** Detach render context resolution from wild resize events, Fix QImage pixel format to match requirements of renderer::ReadFrameBuffer(), set preview widget to be invisible, comment out weird 'background' texture that rendered stripes over the corner of the desired image.
- **Fixed:** Protect CREWaterVolume against crashed due to geometry-less water volumes in Airfield level.
- **Fixed:** Random crash on exit, replace with assert.
- **Fixed:** Pass SSRreflHalfres info into shader to fix bug reading from full-res depth buffer with half res target.
- **Fixed:** Re-enable resizing code in 2D Viewport.
- **Fixed:** Remove unused ColorGrading technique that was giving compiler errors.
- **Fixed:** Allow culling updates even with e\_DebugDraw enabled by removing the check for it in IsStatObjBufferRenderTasksAllowed().
- **Fixed:** Added state garbage collection, fixing assert when DX11 limit of 4096 Blend, DepthStencil, or Rasterizer states was hit.
- **Fixed:** Replace dysfunctional legacy downsample with StretchRegion operation, fixing CD3D9Renderer::CaptureFrameBufferfast(), and thus Statuscope screenshots.
- **Fixed:** Restore context size after generating minimap.
- **Fixed:** Temporarily resize render Viewport during sequence capture to match requested output resolution.
- **Fixed:** Prevent mysterious gamma 'snap back' on quit.
- **Fixed:** Remove unused ColorGrading technique that was giving compiler errors.
- **Fixed:** >0 comparison of uint32 causing GetOrCreateRasterState() failure to crash.
- **Fixed:** Remove hidden dependency of LDR PostAA film grain on the (Dep) Min exposure and (Dep)Max exposure parameters, which, in certain common combinations, would render grain invisible. Now PostAA grain is controlled solely by the grainAmount parameter, reducing confusion and marginally improving performance. The previous behavior can be re-enabled for legacy content support by turning on the CVar r\_GrainEnableExposureThreshold.
- **Fixed:** Remove silly inter-parameter value checks that break the UI, force users to set the oscillator params before setting moveType, and still fail to prevent illegal/illogical states.
- **Fixed:** Pass screensize directly to BlendWeightSMAA\_PS functions, rather than relying on side-effect in global PS\_ScreenSize, which was being modified (illegally w/out DX9 compatibility mode) in main PS function.
- **Fixed:** Preserve all 64 mask bits in MaterialProperties SetFlag() function.
- **Fixed:** Problem in DDS image load function that suppressed INFAs.
- **Fixed:** Silhouettes not getting the render flag to enable depth testing.
- **Fixed:** (Material Editor) Crash when picking material with non .dds texture filename.
- **Fixed:** (Material Editor) Crash when changing textures on material.
- **Fixed:** Memory leak in move assignment of XmlNodeRef.
- **Fixed:** Rendering of brush which have sub objects.
- **Fixed:** Shadow renderviews consuming dirty flag while general renderview was not updating as it was not in the view frustum and was being culled.
- **Fixed:** (Wavicle) New Component naming to properly increment index. Added some effect functions needed for later.
- **Fixed:** (Wavicle) Names of effect templates.
- **Fixed:** (PS4) FogVolumeBoxPS ambiguous call to 'dot' function in the shader.
- **Fixed:** Mouse frame depth test issue.
- **Fixed:** Smart Object rendering issue.
- **Fixed:** Updating aux camera is moved to the latest point so player camera movement will be updated for AuxGeomCBs.
- **Fixed:** AI Debug Info flickering issue.
- **Fixed:** Mouse frame rendering issue.
- **Fixed:** (Entities) RoomscaleCamera: Query HMD's symmetric camera fov.
- **Fixed:** Durango and orbis debug compilation.
- **Fixed:** (Shaders) Message spam due to compilation errors in SkyPS and SkyZPS.
- **Fixed:** GI system freeze during result screen.
- **Fixed:** Crash in GI voxelization.
- **Fixed:** Crash when enabling Total Illumination sometimes causes a crash.
- **Fixed:** (Shaders) SS shadow on grass.
- **Fixed:** (Shaders) too dark SS shadows from the grass.
- **Fixed:** Game Launcher crashes after loading a level with meter per unit set to 0.25.
- **Fixed:** CryPhysX terrain collision.
- **Fixed:** Crash follows assert when painting a terrain layer with proc vegetation.
- **Fixed:** Shadows differ between e\_OnePassOtreeTraversal 0/1.
- **Fixed:** Cubemap generation includes SVOGI contribution.
- **Fixed:** Selected entities not being affected by GI.
- **Fixed:** GI light bounce is culled too early.
- **Fixed:** Assert on loading level particle\_showcase.
- **Tweaked:** (Audio) Wwise added CVar to set panning rule.