Pragma Engine

- Introduction
- Download
- Changelog
- Troubleshooting
- Mapping
- Asset Import / Export
- Console Commands
- Game Mounting
- Developer Documentation
    - Building Pragma
    - Binary Modules

- Git Changelog

Introduction



What is Pragma?

Pragma is a free, open source custom game engine written in C++. Here is an overview over some of its core features:

- Support for various **Source Engine** and **Source 2** assets, such as: maps, models, materials, textures, sound-script files and particle systems
- Extensive modding support with Lua (with Live-coding, debugging-support with the ZeroBrane IDE, support for custom shaders, custom entities/weapons/NPCs/vehicles, etc)
- Support for PBR (**p**hysically **b**ased **r**endering)
- Support for the Cycles-renderer (the same as used by Blender)
- Multiplayer-support, including dedicated servers
- Steam Workshop support
- VR support
- Support for the Bullet and PhysX physics engines
- Support for OpenGL and Vulkan
- AI-system based on behavior trees
- Entity-component-system
- Steam Audio support for spatial audio
- etc.

> Pragma is currently in an early beta, some features may not be fully functional at the moment or are unfinished/experimental.

Pragma also ships with the Pragma Filmmaker, an open source alternative to the Source Filmmaker.

You can use the Book Navigation on the left to navigate to the next page, which has details on how to download and install Pragma.

Click the 

button at the top right for other topics (such as the API documentation for developers).

Media

https://www.youtube-nocookie.com/embed/P7B-_zMSXmA

https://www.youtube-nocookie.com/embed/pJJn6e1sgP0

https://www.youtube-nocookie.com/embed/G1JvibPFnyE

https://www.youtube-nocookie.com/embed/8WvvnuhDgwM

https://www.youtube-nocookie.com/embed/4GRMOU-CdEQ

Download

System requirements

- **Operating System:** Windows 10 (or newer) / Ubuntu 22.04 LTS (or newer)
- **Memory:** 4 GiB RAM or more
- **Graphics card:** GTX 1050 Ti or newer

Download

Standalone

Download

Download the "-win64.zip" version if you're on a Windows OS, and "-lin64.tar.gz" for Linux.
No installation is required, simply extract the archive somewhere and run the "pfm.exe".

Source Code

Pragma and PFM are both completely open-source on GitHub, which means you can also build them yourself. For more information, check out this article.

Changelog

See progress reports for a showcase of new features.

Version 1.0.5 [2022-08-30]

Filmmaker v0.4.6

- Added Pragma Renderer
  - Added SSAA
  - Added motion blur
  - Added directional light maps
  - Added volumetric spot-lights
  - Virtual Reality support
- Added new transform gizmo system
- Added/Improved viewport actor interaction
- Added undo-redo system
- Added decals
- Added tutorials
- Added option for importing map into project
- Added new actor context menu options
- Added support for rendering legacy Eye shader with Cycles X
- General UI improvement updates
- Significantly improve quality of baked lightmaps
- Added components:
  - pfm_baked_lighting
  - pfm_cuboid_bounds
  - pfm_pragma_renderer
  - pfm_motion_blur
  - pfm_rt_mover
  - pfm_overlay_object
  - pfm_camera_actor_link
  - pfm_selection_wireframe
  - pfm_cone_wireframe

Engine

- Added motion blur effect
- Added object BVH system for fast scene intersection tests
- Added directional lightmap baking

- Improved volumetric spot-light effect
- Fixed bloom effect appearing stretched on widescreen resolutions
- Fixed white edges around objects with masked alpha transparency
- Added entity components:
  - renderer_pp_bloom, renderer_pp_dof, renderer_pp_fog, renderer_pp_fxaa, renderer_pp_tone_mapping
  - optical_camera
  - bvh, static_bvh_cache, static_bvh_user
  - light_map_data_cache
- Fixed various crashes

Lua

- Added table.count, table.is_empty
- Added gui.Element:FindAncestorByClass, :SetAutoSizeToContents, :UpdateAutoSizeToContents, :ShouldAutoSizeToContentsX, :ShouldAutoSizeToContentsY
- Added game.Model:GetEyeball, .Load
- Added ents.EyeComponent:FindEyeballIndex, :GetEyeballProjectionVectors
- Added ents.ClickComponent.find_entities_in_kdop
- Added ents.BvhComponent:IntersectionTestKDop
- Added ents.BaseBhvComponent:RebuildBvh, :FindPrimitiveMeshInfo, .HitInfo:CalcHitNormal
- Added ents.DecalComponent:ApplyDecal
- Added util.ImageBuffer:SetPixelColor
- Added game.Model.Mesh.Sub:HasVertexWeights
- Added python library
- Added prosper.util.record_resize_image, .create_generic_image_descriptor_set
- Added Model.Mesh.Sub:MakeVerticesUnique, :SetVertices, :SetIndices
- Added asset.get_asset_state
- Added vector.calc_spherical_stereo_transform
- Added ents.LightMapComponent.DataCache:GetInstanceIds, :GetInstancePose
- Added ents.BaseEnvLightSpotComponent:CalcConeFalloff, :CalcDistanceFalloff
- Added ents.BaseEnvLightPointComponent:CalcDistanceFalloff
- Added util.ThreadPool:WaitForCompletion
- Added ents.citerator, ents.get_all_c, ents.IteratorFilterFunction
- Added classes ents.LightMapDataCacheComponent, ents.LightMapComponent.DataCache

Version 1.0.4 [2022-05-31]

Filmmaker v0.4.5

- Replaced Cycles with Cycles X (With OptiX and OptiX denoising)
- Significantly improved performance of live raytracing render

- Live raytracing render now automatically updates on certain scene changes (e.g. lighting)
- Graph Editor is now mostly functional and can be used for animating
- Added internal Web-Browser which automatically detects and imports downloaded assets
- Added new project file format (old format is no longer supported)
- Added reflection probe and lightmap baking
- Added option to copy/paste and share actors with animation data between projects
- Added secondary and tertiary viewports
- Significantly reduce real-time peformance impact of retargeted actors
- Sky settings have been moved to "sky" actor component to allow animating sky properties
- Sky overrides and angles can now be previewed in the game viewport
- Actors and actor components can now be removed
- Improved asset explorer filter
- "emission_factor" entry field in material editor is now a color field instead of a simple slider
- General load and save performance improvements
- Added quicksave and autosave
- Added new rebindable hotkey system and default hotkeys
- Bones can now be animated with transform widgets in 3D viewport
- Support for volumetric spotlights when rendering with Cycles
- Generated model asset icons now zoom in and focus on the character head for character models
- Added IK configs, IK posing and animating with IK
- Actors can now be right-clicked in game viewport to edit materials directly
- Fixed timeline zoom issues and improved zooming behavior
- Fixed incorrect camera FOV values when importing a SFM session
- Lots of crash, interface, general bug fixes, improvements and QOL changes

Engine

- Added support for meshes with more than 65535 vertices
- Significantly improved asset loading times
- Added model asset import support for formats: blend, fbx, dae, x3d, obj, abc, usd, pmx (MMD)
- Assets can now be automatically detected and imported from archive files
- Added support for Source Engine "Eyes" shader
- Added support for "$alpha" VMT parameter
- Added console commands:
  - asset_multithreading_enabled
  - asset_file_cache_enabled
  - render_enable_verbose_output
  - debug_dump_font_glyph_map
  - locale_localize
  - util_convert_cubemap_to_equirectangular_image
  - util_convert_equirectangular_image_to_cubemap

- Added "-console" launch parameter arguments: "gui", "guid"
- Console command launch parameters that appear after "+map" are now executed after the map has been loaded
- Added entity components:
  - game_animation_player, game_animation_recorder
  - func_portal, portal
  - liquid_surface, liquid_volume, buoyancy, surface, liquid
  - weapon_physcannon
  - wireframe_camera
- Added maps: test_3d_skybox, test_ai_nav, test_fog, test_mirror, test_physics, test_portal, test_water, test_particles
- Added gravity gun script
- Added support for rotating GUI elements
- Added support for UTF8 fonts and localization
- Added "local view target factor" for controlling character eyes
- Updated particle system to new renderer
  - Various blob particle improvements
- Fixed water buoyancy physics
- Fixed fog not rendering
- Fixed 3D skybox not rendering
- Fixed water not rendering
- Fixed issue where light source could appear as turned on despite being turned off or removed previously
- Fixed crash when loading models with custom activity names
- Added Fortress Forever to default mount list
- Added support for equirectangular skyboxes and sky angles
- Fixed crash when attempting to export model asset with invalid flex rules
- Fixed some model meshes not being rendered
- Maps in new-game menu are now split into native maps and external maps
- General performance improvements
- Spot-light inner cone angle has been replaced with blend fraction
- Spot-light outer cone angle has been changed from half-angle to full-angle
- prop_dynamic entities no longer spawn with physics
- General crash fixes and improvements

Lua

- Lua-defined component members can now be animated
- Add "prepared render command buffer" system for issuing multi-threaded render commands
- Component member flags can now be specified as strings
- Added Component member meta data information
- Added function bindings:
  - asset.find, asset.clear_unused, asset.normalize_asset_name, asset.load, asset.reload, asset.precache, asset.is_supported_extension, asset.wait_until_loaded, asset.wait_until_all_pending_jobs_complete, asset.poll, asset.poll_all

- ents.Entity:GetUri, :GetMemberValue, :SetMemberValue, :DebugPrintComponentProperties
- ents.EntityComponent:Log, :GetUri, :GetMemberUri, :GetDynamicMemberIndices, :GetStaticMemberCount, :GetMemberIndices
- ents.ModelComponent:ReloadRenderBufferList, :IsDepthPrepassEnabled, :SetDepthPrepassEnabled, :IsAutoLodEnabled, :SetAutoLodEnabled
- ents.BaseEnvLightSpotComponent:GetBlendFractionProperty, :GetOuterConeAngleProperty, :GetConeStartOffsetProperty
- ents.CameraComponent:GetFocalDistance, :SetFocalDistance, :GetFocalDistanceProperty
- ents.PanimaComponent:DebugPrint, :GetCurrentTime, :SetCurrentTime, :GetCurrentTimeFraction, :SetCurrentTimeFraction, :ReloadAnimation
- ents.LightComponent:SetLightIntensityType, :SetLightIntensity
- ents.ReflectionComponent:GetLocationIdentifier, :GenerateFromEquirectangularImage
- ents.LightComponent:IsBaked, :SetBaked
- ents.add_component_creation_listener
- ents.find_installed_custom_components, ents.get_registered_entity_types
- ents.RenderComponent:IsInPvs
- ents.TransformComponent:Teleport
- ents.WorldComponent:RebuildRenderQueues
- ents.ModelComponent:UpdateRenderMeshes
- file.is_empty
- debug.render_mesh
- game.get_number_of_scenes_queued_for_rendering, game.get_queued_scene_render_info, game.set_gameplay_control_camera, game.reset_gameplay_control_camera, game.get_gameplay_control_camera, game.clear_gameplay_control_camera, game.get_primary_camera_render_mask
- geometry.calc_rect_circle_touching_position
- gui.create_render_target, gui.create_color_image, gui.register_default_skin, gui.set_focus_enabled, gui.is_focus_enabled
- game.DrawSceneInfo:AddSubPass
- gui.Base:CheckPosInBounds
- gui.DropDownMenu:FindOptionIndex
- gui.Element:GetRelativePos, :SetLocalAlpha, :GetLocalAlpha, :IsRemovalScheduled, :ResetRotation, :GetRotationMatrix, :SetRotation, :SetStencilEnabled, :IsStencilEnabled, :SetLocalRenderTransform, :ClearLocalRenderTransform, :DrawToTexture, :GetLocalRenderTransform, :AnchorWithMargin, :ClearFocus, :SetParentAndUpdateWindow
- gui.Text:GetTextHeight, :UpdateSubLines, :SetMaxLineCount, :GetMaxLineCount
- gui.Shape:ClearBuffer, :SetShape, :GetBuffer, :SetBuffer, :GetVertexCount
- gui.WIContextMenu:IsPopulated
- gui.ProgressBar:SetLabelVisible
- input.InputBindingLayer, input.add_input_binding_layer, input.get_input_binding_layers, input.get_input_binding_layer, input.remove_input_binding_layer, input.get_core_input_binding_layers,

    - input.update_effective_input_bindings, input.get_effective_input_bindings, input.set_binding_layer_enabled, input.is_binding_layer_enabled, input.is_binding_layer_enabled, input.InputBindingLayer:FindBoundKeys
    - locale.localize
    - math.snap_to_gridf, math.get_power_of_2_values, math.calc_bezier_point
    - math.Mat4.create_reflection
    - math.Quaternion:MirrorAxis
    - math.Transform:ToPlane, math.Transform:Reflect
    - math.Vector2:GetAngle, math.Vector:Reflect
    - Model:GetExtensionData, :GetSubMesh, :FindSubMeshId
    - Model.Mesh.Sub:GetTriangle, :GetIndexType, :SetIndexType, :AddIndex, :SetIndex, :GetIndex, :HasUvSet, :GetExtensionData, :ReverseIndices, IndexType and MaxIndex enums
    - Renamed Model.Mesh.Sub:GetTriangles to :GetIndices
    - panima.Animation.load, :RemoveChannel
    - panima.Channel:Resize, :GetSize, :SetTime, :SetValue, :ClearValueExpression, :GetValueExpression, :RemoveValue, :FindInterpolationIndices, :FindIndex, :RemoveValueRange, :AddValueRange, :SortValues, :GetInterpolatedValue, :FindIndexRangeInTimeRange
    - prosper.blur_texture
    - prosper.RenderTarget:GetColorAttachmentTexture, :GetDepthStencilAttachmentTexture
    - prosper.Window:GetMonitorBounds, :Maximize, :IsMaximized
    - shader.BaseTexturedLit3D:IsDepthPrepassEnabled, :SetDepthPrepassEnabled
    - table.is_empty, table.table_to_map
    - util.File:GetSize, util.import_assets, util.generate_hair_file, util.generate_hair_data
    - udm.is_numeric_type, is_generic_type, is_non_trivial_type, type_to_string, get_numeric_component, lerp, get_numeric_component_count, is_vector_type, is_matrix_type, get_matrix_row_count, get_matrix_column_count, is_array_type, get_class_type, parse
    - input.is_ctrl_key_down, is_alt_key_down, is_shift_key_down
    - udm.Array:ToTable, udm.PropertyWrapper:ToTable, :ToAscii
    - udm.Property:AddValueRange, :RemoveValueRange, :SetValueType, :RemoveValue, :Copy
    - Material:SetLoaded
- Added classes/enums:
    - util.ThreadPool
    - util.ZipFile
    - gui.DrawToTextureInfo
    - asset.AssetLoadFlags
    - shader.BindState
- Fixed EntityComponent:GetMemberValue, :SetMemberValue not working correctly for some member types
- Fixed util.remove not removing all objects in table correctly in some cases
- Fixed Lua-defined component members of type string not working correctly

Version 1.0.2 [2021-09-13]

Filmmaker v0.4.3

- Added options for animating actor properties
- `util_export_asset` command can now export retargeted models
- Fixed depth buffer issues if camera nearZ or farZ is set to 0
- Fixed some meshes not being rendered when rendering with Cycles/LuxCoreRender

LuxCoreRender

- Added volumetric lighting options
- Added subdivision options for materials
- Added option for enabling PhotonGI cache

Engine

- Added new animation system
  - Added animation math expression system
  - Added animation driver system
- Added entity component member system
- Fixed translucent objects causing depth rendering issues
- Fixed eyes_updown / eyes_rightleft flex controllers not working correctly
- Fixed crash instance when trying to import some Source Engine materials with invalid texture references
- Fixed crash instance when importing some Source Engine maps
- Fixed issue where some meshes would appear invisible if the actor has a skin > 0
- Fixed textures with format ABGR8888 getting imported with swapped color channels

Lua

- Added new Lua API documentation
- Added **CompositeComponent:GetEntities, :GetRootGroup, :AddChildGroup, :GetChildGroups**
- Added **panima** library
- Added **ents.EntityComponent:GetMemberIndex**, **:GetMemberInfo**
- Added **util.get_object_hash**

- Added **ents.UniversalEntityReference**, **ents.UniversalComponentReference** and **ents.UniversalMemberReference**
- Added **ents.AnimatedComponent.parse_component_channel_path**, **:ClearPreviousAnimation**, **:ReloadAnimation**, **:AdvanceAnimations**
- Added **ents.EntityComponent:GetMemberInfos**
- Added **udm.Array:GetArrayType**, **udm.Element:GetArrayValues**, **:SetArrayValues**, **udm.to_json**
- Added type aliases:
  - Entity components can be used in place of entity arguments on C++ function calls
  - EulerAngles and Quaternions can be used interchangeably
  - Vector types can be used interchangeably where applicable

Version 1.0.1 [2021-06-22]

Filmmaker v0.4.2

- Improve retargeting performance
- Fix various retargeting issues
- General retargeting improvements
- Update default material hair settings

Engine

- Added updated Lua documentation and ZeroBrane configuration files
- Updated "lua_help" documentation database
- Added "TF2" and "HL1" demo gamemodes
- Fixed random crashes on some hardware
- Fixed issue where game could freeze permanently during shutdown
- Fixed Lua JIT-Compiler not being enabled
- Fixed importing glTF models saving material files in wrong location
- Fixed exporting maps not working in some cases when the map name has a "." in the name
- Fixed crash on startup with certain launch parameters
- Fixed automatic asset reloading on asset file change not working
- Fixed issue where Steam locations wouldn't get mounted properly on new Steam installations
- Added additional crash dump debug information
- Transitioned AI-nav file-format to UDM system
- Transitioned Lua documentation file-format to UDM system

Lua

- Added **Skeleton:IsRootBone** string parameter overload
- Added **Skeleton:GetBoneHierarchy**
- Added **phys.Transform.IDENTITY**, **:GetAngles**, **:SetAngles**, **:GetForward**, **:GetRight**, **:GetUp**, **.x**, **.y**, **.z**, **.pitch**, **.yaw**, **.roll**, **phys.ScaledTransform.IDENTITY**
- Added **CompositeComponent:ClearEntities**
- Add **EVENT_UPDATE_BONE_POSES** and **EVENT_ON_BONE_POSES_FINALIZED** animated component events
- Added **table.copy**
- Added **engine.library_exists**
- Added **ents.PlayerComponent:SetFlashlightEnabled**, **:ToggleFlashlight**, **:IsFlashlightEnabled**
- General LuaDoc improvements

Version 0.9.15 [2021-05-30]

Filmmaker v0.4.1

- Added console commands: **pfm_max_fps**, **pfm_asset_icon_size**
- Added bone visualization to retarget editor
- Retarget rigs can now be auto-generated from existing rigs if a match can be found
- Various retargeting improvements
- Fixed "hair" settings in material editor not working
- Fixed PFM not showing up in new game menu
- Fixed material editor and explorer not working properly

Engine

- Added console commands: **udm_validate**, **asset_clear_unused**, **asset_clear_unused_textures**, **asset_clear_unused_models**, **asset_clear_unused_materials**, **cache_version_target**, **loc_reload**
- Added launch option: **-audio_api <audioApi>**
- Added game mount priority options
- Physics engine can now be toggled between **PhysX** and **Bullet**
- **Bullet** is now the default physics engine
- Audio engine is now implemented as a module (can be changed in options)
- Assets are now automatically cleared from memory when not in use
- Transition particle system format to UDM
- Fixed entities appearing invisible in some cases
- Fixed various character animation issues

- Fixed character eyes not working correctly if entity is scaled
- Fixed issue where models with a "." in the name could not get imported properly
- Fixed maps not being rendered if map was compiled without BSP
- Fixed crash when generating reflection probes
- Fixed crash when loading map with invalid world model
- Fixed various crash instances
- Fixed various shader/rendering issues

Lua

- Added **debug.beep** and **debug.print**
- Added **phys.CollisionObject:SetAlwaysAwake**, **:IsAlwaysAwake**
- Added **math.map_value_to_fraction** and **math.map_value_to_range**
- Added **locale.set_text**
- Added **composite** and **animated2** entity components
- Added **util.DirectoryChangeListener** class
- Added **animation.Channel**, **.Slice**, **.Animation2**, **.Pose**, **.Player** classes
- Added **ents.Entity:SetUuid**
- Added **ents.RenderComponent:GetRenderModeProperty**
- Added **ents.EyeComponent:GetEyePose**
- Added **udm.PropertyWrapper:GetChildCount**, :**HasValue** and :**ToAscii**
- Added **ents.AnimatedComponent:GetLayeredAnimations**, **:ApplyLayeredAnimations**
- Added **game.Model.Animation:GetBoneId**
- Added **ents.IteratorFilterModel**
- Added **asset.delete**, **.determine_type_from_extension**, **.get_legacy_extension, .get_binary_udm_extension, .get_ascii_udm_extension, .determine_format_from_data, .determine_format_from_filename, .matches, .relative_path_to_absolute_path, .absolute_path_to_relative_path, .get_udm_format_extension, .get_asset_root_directory, .exists, .find_file, .is_loaded**
- Added **file.find_absolute_path**
- Added new overload for **file.remove_file_extension** with parameter to specify which extensions to remove
- Added **udm.AssetData:SetData**
- Added **gui.find_focused_window**, **gui.get_primary_window**, **gui.find_window_under_cursor**, **gui.get_base_elements**, **gui.add_base_element**
- Added **gui.Element:GetRootElement**, **:GetRootWindow**
- Added **shader.BasePbr** base class
- Added **prosper.create_window**, **prosper.WindowCreateInfo** and **prosper.Window** classes
- Fixed **ents.Entity:SetEnabled** Lua binding not behaving correctly in certain cases
- Scripts loaded with **lua_exec** and **lua_exec_cl** now auto reload when changed (Unless"nocache" argument is set)

- **util.register_class** now defines a default constructor and tostring method
- **util.register_class** now returns class object (or nil) instead of boolean

Version 0.9.14 [21-04-16]

Filmmaker

- Added animation and flex retargeting system
- Transform widget can now be toggled between world/local/view space
- Added basic posing and animating capabilities
- Improved hair rendering quality with LuxCoreRender
- Added hair configuration options to material editor
- Added live raytracing preview

Engine

- Added "util_export_asset" option to export assets to Source Engine model/material/texture format with optional fake pbr conversion
- Transitioned asset file formats to UDM system
- Added console commands: "udm_convert", "save", "crash", "debug_render_memory_budget" and "debug_render_memory_stats"
- Exporting model with glTF exporter now includes additional UV sets (such as lightmap UVs)
- Fixed rare random deadlock issue
- Added entity UUIDs to uniquely identify entities
- Added additional crash dump debug information

Lua

- Added **asset.get_supported_extensions, asset.export_texture_as_vtf, asset.TYPE_MAP**
- Added **ents.GetUuid, ents.AttachableComponent:GetBone**
- Added **Quaternion:Distance**
- Added class **ents.IteratorFilterUuid**
- Added **udm** library
- Added **game.get_material**
- Added **util.BSPTree.Node:GetIndex**
- Added **game.Material:Reset**
- Added **game.Model.Animation.Load**, **game.Model.FlexAnimation.Load**
- Added **game.Model.Mesh.Sub:Copy**
- Added **game.Model.Animation:GetBoneId**
- Added **game.Model.Eyeball:GetUpperLidFlexIndices**, **:GetUpperLidFlexAngles**, **:GetLowerLidFlexIndices**, **:GetLowerLidFlexAngles**, **:GetUpperLidFlexIndex** and

:GetLowerLidFlexIndex
- Added **game.Model.FCOPY_BIT_FLEX_ANIMATIONS** flag
- Added **ents.ModelComponent:GetBodyGroups**, **:SetBodyGroups**
- Added **ents.AnimatedComponent:GetBaseAnimationFlags**, **:SetBaseAnimationFlags**, **:GetLayeredAnimationFlags**, **:SetLayeredAnimationFlags, ents.AnimatedComponent:SetAnimatedRootPoseTransformEnabled, :IsAnimatedRootPoseTransformEnabled, :AddRootPoseBone**, **:SetRootPoseBoneId**, **:GetRootPoseBoneId** and **ents.AnimatedComponent.FPLAYANIM_LOOP**
- Added **ents.LightComponent:SetMorphTargetsInShadowsEnabled**, **:AreMorphTargetsInShadowsEnabled, .BakeSettings.globalLightIntensityFactor** and **.import_lightmap_atlas**
- Added **ents.ReflectionProbeComponent:GetIBLMaterialFilePath**, **:CaptureIBLReflectionsFromScene** and **:RequiresRebuild**
- Added **prosper.ImageCreateInfo:FLAG_ALLOCATE_DISCRETE_MEMORY_BIT**, **FLAG_DONT_ALLOCATE_MEMORY_BIT**, **FLAG_SRGB_BIT**, **FLAG_NORMAL_MAP_BIT**
- Added **prosper.Image:IsSrgb**, **:IsNormalMap**, **:SetSrgb**, **:SetNormalMap** and **:Convert**
- Added **prosper.Mesh:ClearBuffers**
- Added **ents.LightMapComponent:SetLightmapAtlas**
- Added **ents.RenderComponent:ClearBuffers**
- Removed **game.Model.Eyeball:GetUpperFlexDesc**, **:GetLowerFlexDesc**, **:GetUpperTarget**, **:GetLowerTarget**, **.lowerLidFlexDesc** and **.upperLidFlexDesc**

Version 0.9.13 [2021-03-14]

Filmmaker

- Fully integrated LuxCoreRender
- Added support for transparent backgrounds/skies when rendering with LuxCoreRender
- Added new render tool commands (LuxCoreRender only):
  - preview: Creates a preview image of what's been rendered so far and opens it in the default image program
  - pause: Pauses the current render
  - resume: Resumes the current render
  - stop: Stops the render and uses what's been rendered so far to generate the final image
  - suspend: Creates a .rsm-file, which can be used to continue the render at a later date
  - export: Exports the entire scene in a LuxCoreRender format (including models and textures), which can be used to render it with the standalone LuxCoreRender

program
- Fixed issue where importing textures via drag-and-drop would not generate mipmaps

Engine

- Updated build instructions
- Morph targets now affect shadows
- Increased bone limit from 512 to 1024
- Added console commands "strip_weapons" and "cl_render_frustum_culling_enabled"
- Reduced number of entities with unnecessary tick/logic events to improve performance
- Characters now blink if no eyeballs are defined in the model, but blink flex controllers exist
- Added UDM data format for unifying asset file formats
- Fixed issue where viewmodel entities would use incorrect field of view
- Fixed viewmodel not appearing in some cases if viewmodel was changed after weapon was deployed
- Changed entity component net event system so net events no longer have to be registered on the server *before* they are registered on the client
- Merged cengine with client library and engine with shared

Lua

- Removed gamemode system and replaced it with gamemode entity components
  - Allows changing gamemodes on the fly, as well as running multiple different gamemodes at the same time
- Added "player" and "gamemode" entity component types (and sub-directories)
- All entity components can now have individual tick/logic events instead of relying on the logic component
- Added **udm** library for loading/saving/working with udm assets
- Added **ents.ModelComponent:GetBodyGroups** and **:SetBodyGroups**
- Added **math.Quaternion:Distance**
- Added **ents.PhysicsComponent:EnableCollisions**, **:DisableCollisions**, **:SetCollisionsEnabled**, **:SetSimulationEnabled** and **:IsSimulationEnabled**
- Added **ents.CharacterComponent:SetMoveController** and **:GetMoveController**
- Added **util.DataStream:WriteVertex**, **:ReadVertex**
- Added **Vector:GetYaw** and **:GetPitch**
- Added **util.generate_uuid_v4**
- Added **ents.register_component_net_event**
- Added **ents.AttachableComponent.FATTACHMENT_MODE_FORCE_TRANSLATION_IN_PLACE**, **FATTACHMENT_MODE_FORCE_ROTATION_IN_PLACE** and **FATTACHMENT_MODE_FORCE_IN_PLACE**

Version 0.9.12 [2021-02-08]

Filmmaker

- Added support for rendering with **LuxCoreRender**
- Added fur rendering options with LuxCoreRender

Engine

- Massive general performance improvements
- Added automatic rendering instancing system
- Added flex animation system
- Added new bloom/glow implementation
- Added lightmap baking with Cycles
- Added console commands **debug_render_octree_dynamic_find**, **debug_render_stats**, **render_vsync_enabled**, **render_draw_static**, **render_draw_dynamic**, **render_translucent**, **cl_render_shadow_lod_bias**, **render_queue_worker_thread_count**, **debug_textures**, **sky_override**, **render_queue_entities_per_worker_job**, **render_queue_worker_jobs_per_batch**, **render_instancing_threshold** and **render_instancing_enabled**
- Fixed issue where camera would move upwards continuously in fullscreen windowed resolution mode
- Fixed first LOD being skipped when importing Source Engine models
- Fixed model LODs not working
- Fixed various crash bugs
- Fixed bone scaling not working correctly in some cases

Lua

- Added **ents.RenderComponent:GetLocalRenderBounds**, **ents.RenderComponent:GetLocalRenderSphereBounds**, **ents.RenderComponent:GetAbsoluteRenderBounds**, **ents.RenderComponent:GetAbsoluteRenderSphereBounds**, **ents.RenderComponent:SetLocalRenderBounds**, **ents.RenderComponent:ShouldCastShadows**, **ents.RenderComponent:ShouldDraw**, **ents.RenderComponent:ShouldDrawShadow**, **ents.RenderComponent.EVENT_ON_RENDER_MODE_CHANGED** and **ents.RenderComponent.EVENT_UPDATE_INSTANTIABILITY**
- Added **util.DataBlock:GetVector2**
- Added **game.Material.detail_blend_mode_to_enum** and **game.Material.DetailBlendMode** enums
- Added **ents.SceneComponent:GetRenderQueue**, **ents.SceneComponent:BuildRenderQueue** and **ents.SceneComponent.CreateInfo**
- Added **math.Vector:ToScreenUv**

- Added **game.set_default_game_render_enabled**, **game.is_default_game_render_enabled**, **game.render_scenes**, **game.set_debug_render_filter** and **game.update_render_buffers**
- Added **game.RENDER_FLAG_TRANSLUCENT_BIT**
- Added **util.ImageBuffer:SwapChannels**
- Added **prosper.get_api_identifier**, **prosper.get_api_abbreviation**, **prosper.wait_for_current_swapchain_command_buffer_completion**, **prosper.create_primary_command_buffer** and **prosper.create_secondary_command_buffer**
- Added **prosper.CommandBuffer:Flush**
- Added **game.DrawSceneInfo.FLAG_FLIP_VERTICALLY_BIT**, **game.DrawSceneInfo.FLAG_DISABLE_RENDER_BIT** and **game.DrawSceneInfo.flags**
- Added classes **game.RenderQueue**, **game.DepthStageRenderProcessor**, **game.LightingStageRenderProcessor**, **shader.GameWorld**, **game.Model.FlexAnimation** and **game.Model.FlexAnimation.Frame**
- Added **ents.ModelComponent:SetMaxDrawDistance**, **ents.ModelComponent:GetMaxDrawDistance** and **ents.ModelComponent.EVENT_ON_RENDER_MESHES_UPDATED**
- Added **ents.AnimatedComponent:AreSkeletonUpdateCallbacksEnabled**, **ents.AnimatedComponent:SetSkeletonUpdateCallbacksEnabled**, **ents.AnimatedComponent.EVENT_ON_SKELETON_UPDATED**, **ents.AnimatedComponent.EVENT_ON_BONE_MATRICES_UPDATED** and **ents.AnimatedComponent.EVENT_ON_BONE_BUFFER_INITIALIZED**
- Added **util.BSPTree:FindLeafNodesInAABB**
- Added **ents.LightmapComponent.bake_lightmaps**, **ents.LightmapComponent:SetExposure**, **ents.LightmapComponent:GetExposure**, **ents.LightmapComponent:GetExposureProperty** and **ents.LightmapComponent.BakeSettings**
- Added **ents.FlexComponent:PlayFlexAnimation**, **ents.FlexComponent:StopFlexAnimation**, **ents.FlexComponent:GetFlexAnimationCycle**, **ents.FlexComponent:SetFlexAnimationCycle** and **ents.FlexComponent:SetFlexAnimationPlaybackRate**
- Added **game.Model:GetFlexAnimations**, **game.Model:GetFlexAnimationNames**, **game.Model:GetFlexAnimationCount**, **game.Model:AddFlexAnimation**, **game.Model:LookupFlexAnimation**, **game.Model:GetFlexAnimation**, **game.Model:GetFlexAnimationName**, **game.Model:ClearFlexAnimations**, **game.Model:RemoveFlexAnimation**, **game.Model.AddFlexControllerId**, **game.Model.SetFlexControllerValue**, **game.Model.LookupLocalFlexControllerIndex**, **game.Model:GetLOD** and **game.Model.Mesh:RemoveSubMesh**
- Added **game.RasterizationRenderer:ReloadPresentationRenderTarget**
- Added **file.to_relative_path**
- Added **geometry.calc_point_on_triangle** and **geometry.calc_triangle_area**
- **util.save_image** now has an option for saving cubemap textures

- Removed **ents.SceneComponent:PrepareRender**, **ents.TransformComponent:GetPosProperty**, **ents.TransformComponent:GetRotationProperty**, **ents.TransformComponent:GetScaleProperty**, **game.draw_scene**, **ents.RenderComponent:GetRenderBounds**, **ents.RenderComponent:SetRenderBounds**, **ents.RenderComponent:GetRenderSphereBounds**, **ents.RenderComponent:GetRenderModeProperty**, **ents.RenderComponent:SetDepthBias, ents.RenderComponent:GetDepthBias**, **game.DrawSceneInfo.flipVertically** and **ents.RenderComponent.RENDERMODE_AUTO**
- Changed parameters of **ents.SceneComponent:Render** and **ents.RenderComponent:UpdateRenderBuffers**

Version 0.9.11 [2020-11-11]

Filmmaker

- Cycles render settings are now saved with the project
- Changed frame render order when rendering animations with external render tool for easier quality control
- Added support for detail maps when rendering with Cycles
- Fixed emission textures having no effect when rendering with Cycles
- Fixed IOR value being set improperly when opening material in material editor
- Fixed material editor not updating properly when switching to a different material
- Fixed HDR output format not working properly when rendering with Cycles
- Changed default gamma correction value when rendering with Cycles to 2.2 (was 2.4 before)

Engine

- Added support for **$detail**, **$detailscale**, **$detailblendfactor**, **$detailblendmode** and **$detailtint** VMT parameters
- Added CS: GO to default game mount list
- Fixed issue where entities with morph target animations would get visually corrupted if they were made invisible and then visible again
- Fixed issue where ingame cursor position would mismatch actual cursor position in fullscreen windowed resolution mode
- Fixed issue where camera would move upwards continuously in fullscreen windowed resolution mode

Lua

- Replaced class **game.Scene** with **ents.SceneComponent**
- Added **ents.SceneComponent.OcclusionCullingMethod** enums
- Added **game.Material.DetailBlendMode** enums
- Added **game.Material.detail_blend_mode_to_enum**
- Added **ents.SceneComponent:SetOcclusionCullingMethod**
- Added **game.Model:GetBodyGroupMesh**
- Added **game.Material:SetShader**
- Added **ents.Entity:GetBodyGroup** and **:SetBodyGroup** overloads for body group IDs
- Added **ents.BaseEntityComponent:Save**, **ents.BaseEntityComponent:Load** and **ents.BaseEntityComponent:Copy**
- Added **ents.Entity:RemoveFromAllScenes** and **ents.Entity:AddChild**
- Added **debug.breakpoint** (Only available if `-luaext` was specified in the launch options)
- Added **math.Mat4:ApplyProjectionDepthBiasOffset**
- Added **ents.RenderComponent:SetReceiveShadows** and **ents.RenderComponent:IsReceivingShadows**
- Added **ents.EyeComponent:ClearViewTarget**
- Added **util.DataBlock:GetVector2**
- Renamed **game.create_scene** to **ents.create_scene**
- **util.remove** now has a second argument for removing objects safely (if available)
- **toboolean** and **toint** now accept numbers and booleans as arguments
- Changed parameter order of **ents.create_prop**

Version 0.9.10

Filmmaker

- Added Cycles option for setting the number of frames to the end of a clip or session
- Render jobs now take up a lot less disk space for animations (static geometry is now stored shared instead of every frame)
- Slightly improved overall rendering speed
- Fixed rendering taking up your entire CPU resources when rendering with your GPU
- Fixed an issue where rendering several times consecutively could slow your PC down to a crawl
- Fixed VR renders being flipped horizontally
- Fixed a denoising issue which caused less than optimal denoising quality
- Fixed issue where the tiles in the final image would have mismatching sample counts and some tiles would be lower quality as a result
- Fixed render preview not being interactable when rendering VR images
- Fixed potential crash when doing a preview render with the Vulkan renderer being used

Engine

- Minor occlusion culling performance optimizations

- Greatly improved reflection probe rendering speed
- Fixed issue where entities would get rendered with error materials if their skin would exceed the max skin count of their model
- Fixed bone scaling not working correctly
- Fixed massive performance hit for light sources that cast shadows in the past, but had shadow casting disabled
- Fixed BSP occlusion culling not working for imported Source Engine maps
- Fixed meshes not being rendered if entity skin exceeds maximum skin count
- Improved lightmap implementation
- Various minor bug fixes and improvements
- Improved loading times when starting a game

Lua

- Added class **util.Version**
- Added **exec**
- Added **ents.create_prop**
- Added **ents.AnimatedComponent:SetBindPose** and **ents.AnimatedComponent:GetBindPose**
- Added **ents.LightMapComponent:UpdateLightmapUvBuffers**, **ents.LightMapComponent:ReloadLightmapData** and **ents.LightMapComponent:SetLightmapAtlas**
- Added **ents.LightMapReceiverComponent:UpdateLightmapUvData**
- Added **gui.Element:InjectMouseClick** and **gui.Element:InjectKeyPress**
- Added **ents.Entity:IsDisabled** and **ents.Entity:IsTurnedOff**
- Added **util.Path:CreateFilePath** and **util.Path:CreatePath**
- Added **ents.MapComponent:GetMapIndex**
- Added **ents.AnimatedComponent:PlayAnimation** overloads for playing animations by id
- Added **util.DataBlock:ToString**
- Added **game.Model.Frame:Copy**, **game.Model.Skeleton.Bone:IsAncestorOf** and **game.Model.Skeleton.Bone:IsDescendantOf**
- Added **math.calc_average_rotation**
- Added **file.strip_illegal_filename_characters**
- Added **game.Model.Mesh:GetReferenceId**
- Added **game.Model.Mesh.Sub:FlipTriangleWindingOrder**, **:SetVertexCount**, **:SetIndexCount**, **:AddUVSet, :GetUVSetNames**, **:SetVertexTangent** and **:HasUVSet**
- Added **string.hash**
- Added new overload for **game.Model.Mesh.Sub:GetUVs**, **:SetVertexUV**, **:GetVertexUV** with UV set parameter
- Changed argument format for **util.pack_zip_archive**
- Renamed **ents.AnimatedComponent:GetBindPose** to **ents.AnimatedComponent:GetBoneBindPose**
- **util.remove** now accepts a table as argument
- **file.write** now automatically creates the file path if it doesn't exist
- File write operations now accept paths that include "addons/<addonName>/" as prefix

Troubleshooting

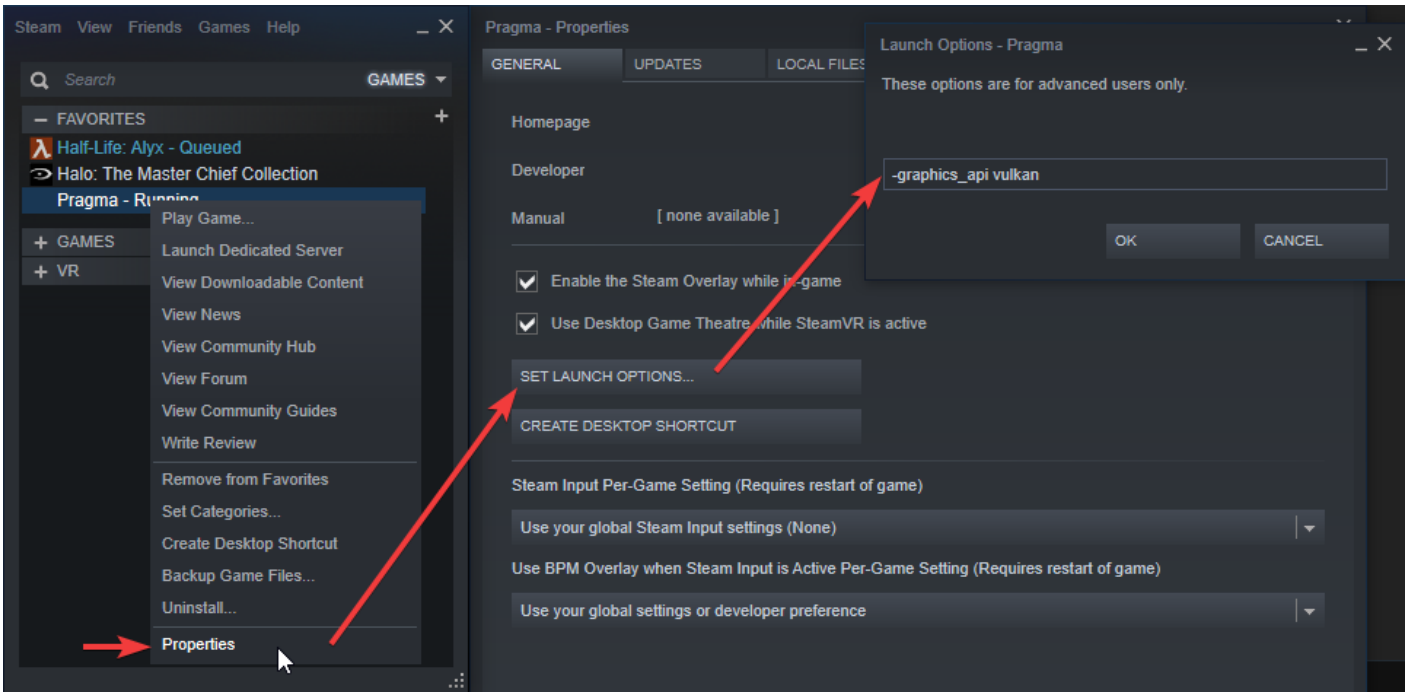
Pragma crashes on startup

If Pragma crashes on startup, please ensure that your system fulfills the system requirements and that your graphics drivers are up to date.

If you have multiple GPUs, please make sure to enable your high-default GPU as the default for Pragma. You can follow the instructions on this page to do so.

There is no text in the main menu

Try adding `-graphics_api vulkan` to the launch options in Pragma to switch to the Vulkan renderer:



I have an AMD GPU and Pragma/PFM crashes on startup

This should be fixed in newer versions of PFM, until they are released publicly, you can try this temporary fix:

1. Go to the Pragma root folder
2. Remove the `cache` directory
3. Go to `Pragma/shaders/world/eye/`
4. Rename `fs_eye.gls` to `fs_eye_bak.gls`
5. Launch PFM

Character eyes won't be rendered properly in the real-time render viewport, but it shouldn't be freezing anymore.

Mapping

Asset Import / Export

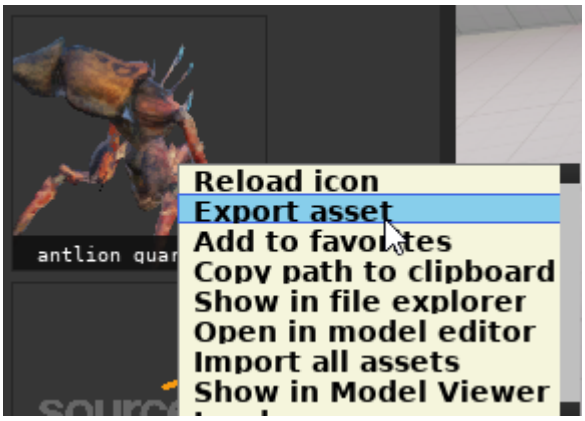
Export

Pragma comes with the following export capabilities:

- You can use Pragma to export Source Engine and Source 2 Engine assets for use in modelling programs
- Assets in Pragma can be exported in the Source Engine model/material/texture format
  - Fake PBR can be applied automatically for use in SFM
- Automatic generation of ambient occlusion maps (if they don't exist)
- Automatic re-scaling of the models to meters
- Conversion to match Blender's coordinate system
- Includes the rig, all of the animations and morph targets of the model
- Conversion to standard normal maps (e.g. if self-shadowed bumpmaps are used)
- Conversion of Source 1 assets with PBR settings
- Automatic conversion of the textures to png, bmp, tga, jpg, hdr, dds, ktx or vtf
- Automatic conversion of the textures to a RMA format
- Works with maps / level geometry

Exporting Assets

If you want to export a model or material, the easiest way to do so is via the Filmmaker interface. Simply open the model/material catalog, right click on the model icon, then click "Export asset":



If you want to export a different asset type, or need more control, you can use the export command utility instead. To use it, start a game in Pragma first. You can either load a map (which map doesn't matter), or run "map empty" in the console. Once the map has been loaded, you can use the "util_export_asset" console command with the following parameters:

- **-model** <*modelName*>: Exports the specified model. If a directory is specified, all models within that directory will be exported.
- **-texture** <*textureName*>: Exports the specified texture(s). If a directory is specified, all textures within that directory will be exported.
- **-material** <*materialName*>: Exports the specified material(s). If a directory is specified, all materials within that directory will be exported.
- **-map** <*mapName*>: Exports the specified map(s). If a directory is specified, all maps within that directory will be exported.
- **-export_images** *1/0*: If enabled, all textures of the asset will be converted to the specified image output format and saved alongside the asset. Default: 1
- **-normalize_texture_names** *1/0*: If enabled, the exported textures will be renamed according to their type (e.g. "textureName_normal") and will not keep their original name. Default: 0
- **-image_format** *png/bmp/tga/jpg/hdr/dds/ktx*: The format to use for exported textures. Default: dds
- **-binary** *1/0*: If enabled, the model will be exported in the "glb" format, otherwise "glTF". Default: 0
- **-verbose** *1/0*: If enabled, additional information will be printed to the console during the export. Default: 1
- **-recursive** *1/0*: If enabled and a directory is specified as the asset, all assets from all sub-directories will be exported as well. Default: 0

Models only:

- **-export_animations** *1/0*: If enabled, all animations of the model will be exported as well. This may drastically increase the file size and loading times when importing the model into a modelling program. Default: 1
- **-export_morph_targets** *1/0*: If enabled, will export all morph targets (flexes) of the model as well. Default: 1
- **-export_skinned_mesh_data** *1/0*: If disabled, no skeleton, vertex weights or animations will be exported. Default: 1
- **-embed_animations** *1/0*: If enabled, all of the animations of the model will be embedded in the exported glTF file. When disabled, each animation will be exported as a separate file instead. Default: 1
- **-full_export** *1/0*: If enabled, body groups of the model will be exported as well. Default: 0
- **-scale** <*scale*>: The value by which to scale the exported meshes. Use 1 to keep the original scale. Use 0.025 to convert the meshes to meters. Default: 0.025
- **-generate_ao** *1/0*: If enabled, will automatically generate ambient occlusion maps for the model if it doesn't have any. Default: 0
- **-ao_resolution** <*resolution*>: The resolution to use for the ambient occlusion maps if "generate_ao" is enabled. Default: 512
- **-ao_samples** <*samples*>: The number of samples to use for generating the ambient occlusion maps if "generate_ao" is enabled. Default: 40
- -**ao_device** *cpu/gpu*: Whether to use the CPU or GPU to generate the ambient occlusion maps if "generate_ao" is enabled. Default: cpu

- **-merge_meshes_by_material** *1/0*: If enabled, all meshes that use the same material will be merged. Default: 1
- **-enable_extended_dds** *1/0*: If enabled and the image format is set to dds, will enable support for BC6 and BC7 compression using the DXT10 version of the format. Default: 0
- **-list_animations** *1/0*: If enabled, the model will not be exported and all available animations of the model will be listed in the console instead. Default: 0
- **-animation** *<animName>*: If specified, only this animation will be exported.
- **-format** *glTF/mdl*: The output format for the model. If "mdl" is specified TODO
- **-game** *1/0*: Only has an effect if "format" is set to "mdl". TODO
- **-preview** *1/0*: If enabled, the model will be shown in the viewport. Default: 0

> If you want to use Pragma to export Source Engine assets, you generally don't have to copy any of the assets to Pragma, the Engine should be able to locate the assets automatically as long as they're part of one of your installed Source Engine games on Steam.

Usage Examples

Exporting a material (including textures):

```
util_export_asset -material "brick/brick_1a" -verbose -image_format png
```

Exporting a texture:

```
util_export_asset -texture "concrete/concrete_5_floor_3_psd_918fcd2d" -verbose -image_format
png
```

Exporting a map:

```
util_export_asset -map "gm_construct" -verbose -export_images 1 -image_format png -generate_ao
0 -export_animations 0
```

Exporting a model:

```
util_export_asset -model "props_2fort/tank001" -verbose -image_format png -generate_ao 0
```

(Make sure to remove the ".mdl" extension!)

Exporting a Source 2 model:

```
util_export_asset -model "characters/gman/gman" -verbose -image_format png -generate_ao 0
```

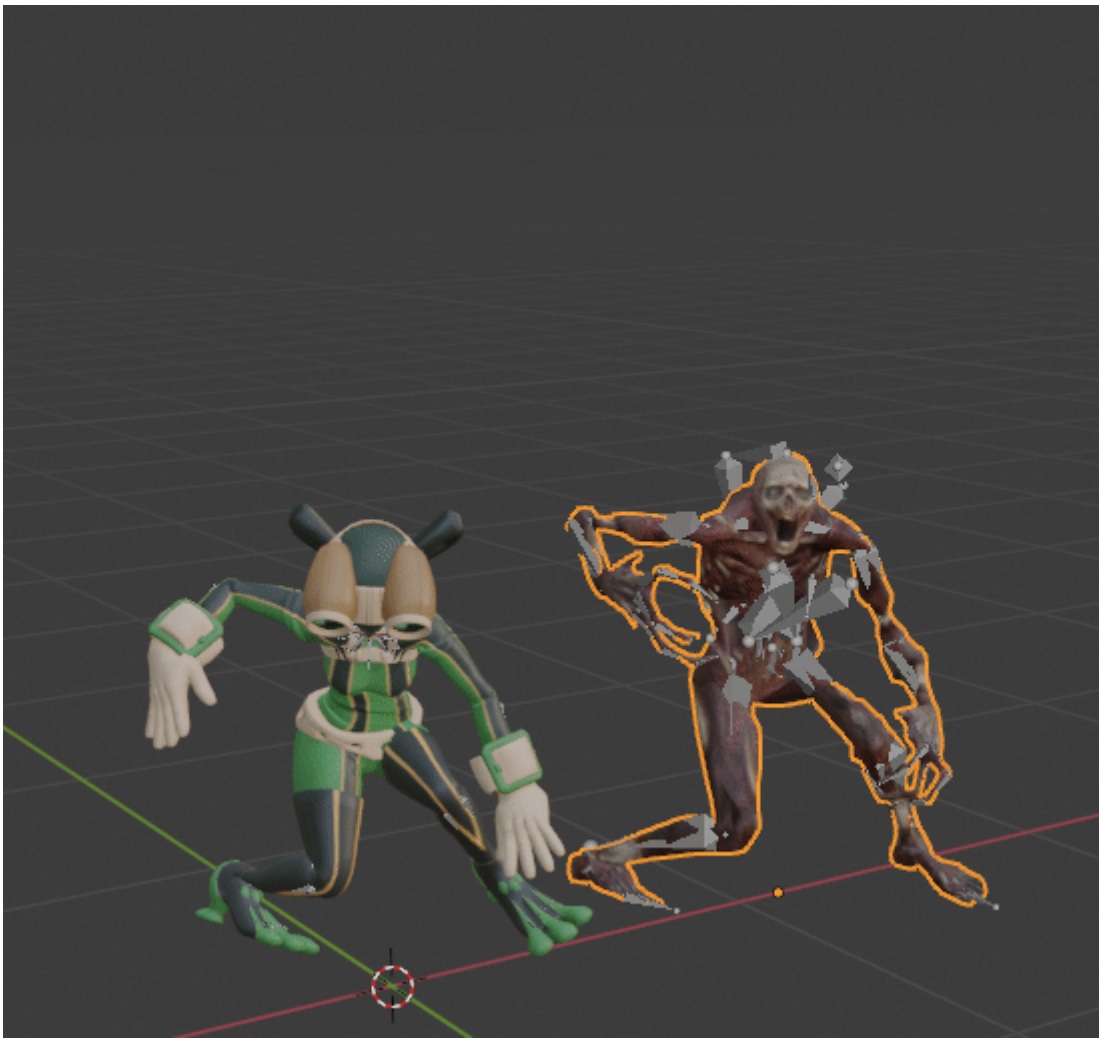
Batch-exporting all models in a directory and all sub-directories:

```
util_export_asset -model "props_2fort" -verbose -image_format png -generate_ao 0 -recursive 1
```

Exporting retargeted models

> This section refers to PFM v0.4.3 and newer and may not be representative of older versions.

It is also possible to export a model with retargeted animations (i.e. export a model with the animations of another model.):



To do so, you will have to set up a retarget rig in PFM's retarget editor first:

https://www.youtube-nocookie.com/embed/2f-pxcjbanA

Once you have created the retarget rig, you can use the `-retarget_source` argument in the `util_export_asset` command to export the model with the retargeted animations:

```
util_export_asset -model "e3assassin" -retarget_source "re5/jv1" -verbose -image_format tga -generate_ao 0 -embed_animations 1
```

Exporting to Source Engine

You can also use the command to export a model **to** the Source Engine .mdl format. To do so, simply set the "format" parameter to "mdl":

```
util_export_asset -model "props_2fort/tank001" -format "mdl"
```

> Most of the parameters listed above will have no effect when using the Source Engine mdl-exporter!

This will automatically generate the QC and SMD files for the model, convert the textures to vtf, convert the materials to vmt, as well as automatically compile the model using studiomdl. Pragma will attempt to locate studiomdl automatically, but you can also specify which game's studiomdl should be used:

```
util_export_asset -model "props_2fort/tank001" -format "mdl" -game "tf2"
```

This will use the studiomdl from TF2. The game name has to match the entry from "Pragma/cfg/mounted_games.udm" (You can open the file in a text-editor).

If the model's materials use PBR, the exporter will also automatically apply fake PBR for use in SFM.

All of the model's animations will be included in the Source Engine model as well, however morph targets are currently not supported.

Using Pragma to convert models from Blender to Source

Converting a model from Blender (or other modelling programs) is very simple. All you have to do is export your model from Blender in either the glTF format, or the glb format and copy the exported file to "Pragma/addons/imported/models". Then use the command above and replace the model name with the name of the glTF/glb file (without the extension).

Source 2

The approach for Source 2 assets is exactly the same as for Source 1 assets, with the exception of maps. To export a Source 2 map, follow these steps (using Half-Life: Alyx as an example):

1. Open the map's vpk-file (steamapps/common/Half-Life Alyx/game/hlvr/maps) in GCFScape
2. Extract all of the files from the vpk to "Pragma/addons/imported/"
3. Run the "util_export_asset" command with the asset name pointing to the "vwrld_c"-file. For example, for the map "a5_vault", you should use:

```
util_export_asset -map "a5_vault/world" -verbose -export_images 1 -image_format png -generate_ao 0
```

Import

If you want to import a model from Blender (or other modelling programs), all you have to do is export the model in the glTF or glb format and copy the exported file to "Pragma/addons/imported/models". The model should automatically show up in the PFM model catalog and it will automatically get converted (along with the textures) when used.

Source Engine / Source 2

If you want to import a model (or material/texture) from Source or Source 2, you generally don't have to do anything. All of your installed Source Engine games (assuming they're installed via Steam) should be automatically detected and mounted by Pragma. In the rare event that this doesn't work, you can try adding the mount path for the game to "Pragma/cfg/mounted_games.udm".

Alternatively you can also simply extract the asset files to "Pragma/addons/imported/models" and "Pragma/addons/imported/materials" respectively.

Console Commands

Variables

audio_api <audioApi>

Type: String

The underlying audio API to use.

cache_version <value>

Type: String

The engine version that the cache files are associated with. If this version doesn't match the current engine version, the cache will be cleared.

cache_version_target <value>

Type: String

If cache_version does not match this value, the cache files will be cleared and it will be set to it.

cl_allowdownload <value>

Type: String

cl_allowupload <value>

Type: String

cl_audio_always_play <value>

Type: String

0 = Don't play sounds if window isn't focused., 1 = Always play sounds

cl_audio_hrtf_enabled <value>

Type: String

Enables or disables Head-related transfer function.

cl_audio_master_volume <value>

Type: String

Volume scale for all sounds.

cl_audio_streaming_enabled <value>

Type: String

0 = All sounds will be loaded immediately (= Slower load times), 1 = Some sounds will be loaded over time. (= Sounds might start with a delay)

cl_controller_axis_input_threshold <value>

Type: String

Axis inputs below this threshold will be ignored for regular console commands.

cl_controller_enabled <value>

Type: String

Enables or disables game controllers / joysticks.

cl_downscale_imported_high_resolution_rma_textures 1/0

Type: Boolean

If enabled, imported high-resolution RMA textures will be downscaled to a more memory-friendly size.

cl_effects_volume <value>

Type: String

Volume scale for effect sounds (e.g. footsteps, gunshots, explosions, etc.).

cl_flex_phoneme_drag <value>

Type: String

Changes the time required for facial phoneme flexes to adapt to new phonemes.

cl_fov <value>

Type: String

Horizontal world field of view.

cl_fov_viewmodel <value>

Type: String

Horizontal view model field of view.

cl_fps_decay_factor <value>

Type: String

How slowly to decay the previous fps.

cl_gpu_device <value>

Type: String

The unique identifier for the GPU vendor and device to use. If empty (or invalid), the first detected device will be used. Usage: <vendorId,deviceId>

cl_gpu_timer_queries_enabled <value>

Type: String

Enables or disables GPU timer queries for debugging.

cl_gui_volume <value>

Type: String

Volume scale for sounds emitted by GUI elements.

cl_language <value>

Type: String

Game language.

cl_material_streaming_enabled <value>

Type: String

0 = All materials and textures will be loaded immediately (= Slower load times), 1 = All materials and textures will be loaded over time. (= Black textures until loaded)

cl_max_fps <value>

Type: String

FPS will be clamped at this value. A value of < 0 deactivates the limit.

cl_mouse_acceleration <value>

Type: String

Amount of mouse acceleration. 1 = No acceleration.

cl_mouse_pitch <value>

Type: String

Mouse sensitivity on the pitch axis. Use positive values to invert the axis.

cl_mouse_sensitivity <value>

Type: String

Scale for the mouse sensitivity.

cl_mouse_yaw <value>

Type: String

Mouse sensitivity on the yaw axis.

cl_music_volume <value>

Type: String

Volume scale for music.

cl_physics_simulation_enabled <value>

Type: String

Enables or disables physics simulation.

cl_port_tcp <value>

Type: String

Port used for TCP transmissions.

cl_port_udp <value>

Type: String

Port used for UDP transmissions.

cl_render_anti_aliasing <value>

Type: String

0 = No Anti Aliasing, 1 = MSAA, 2 = FXAA.

cl_render_brightness <value>

Type: String

Scene brightness.

cl_render_contrast <value>

Type: String

Scene contrast.

cl_render_depth_of_field <value>

Type: String

Depth of field scale.

cl_render_fov <value>

Type: String

Specifies the horizontal field of view.

cl_render_frustum_culling_enabled <value>

Type: String

Enables or disables frustum culling.

cl_render_fxaa_edge_threshold <value>

Type: String

The minimum amount of local contrast required to apply algorithm.

cl_render_fxaa_min_edge_threshold <value>

Type: String

Trims the algorithm from processing darks.

cl_render_fxaa_sub_pixel_aliasing_removal_amount <value>

Type: String

The amount of sub-pixel aliasing removal.

cl_render_lod_bias <value>

Type: String

Model LOD-bias. Higher values means higher LOD-Models will be used at lower distance (=Lower quality).

cl_render_monitor <value>

Type: String

Which monitor to use in fullscreen mode.

cl_render_motion_blur <value>

Type: String

Amount of motion blur. 0 = disabled.

cl_render_msaa_samples <value>

Type: String

1 = MSAAx2, 2 = MSAAx4, 3 = MSAAx8, etc.

cl_render_occlusion_culling <value>

Type: String

0 = Off, 1 = Brute-Force, 2 = CHC++, 3 = Octree, 4 = BSP +Octree

cl_render_particle_quality <value>

Type: String

Quality of particle systems. 0 = No particles will be rendered, 1 = Particles will be unlit, 2 = Particles will receive lighting, 3 = Particles will cast shadows.

cl_render_present_mode <value>

Type: String

0 = Immediate, 1 = FIFO, 2 = Mailbox

cl_render_preset <value>

Type: String

Selected preset in display options.

cl_render_reflection_quality <value>

Type: String

Changes the quality of reflections. 0 = Only static reflections, 1 = Dynamic reflections, 2 = Reflections with particle effects.

cl_render_resolution <value>

Type: String

Changes the internal rendering resolution. If left empty, the resolution will be the same as the window resolution!

cl_render_shader_quality <value>

Type: String

Shader quality. The actual effect depends on the shader. 1 = Lowest Quality, 10 = Highest Quality

cl_render_shadow_dynamic <value>

Type: String

Turns dynamic shadows on or off.

cl_render_shadow_lod_bias <value>

Type: String

Model LOD-bias for shadows. Higher values means higher LOD-Models will be used at lower distance (=Lower quality). This value is added on top of cl_render_lod_bias!

cl_render_shadow_pssm_split_count <value>

Type: String

The number of cascades to be used for PSSM. Cannot be 0 or higher than 4.

cl_render_shadow_pssm_update_frequency_offset <value>

Type: String

Update frequency for PSSM shadows in frames, relative to 'cl_render_shadow_update_frequency'.

cl_render_shadow_resolution <value>

Type: String

Shadowmap Resolution. Higher resolution equals higher quality shadows, but is also more expensive to render.

cl_render_shadow_update_frequency <value>

Type: String

Update frequency in frames. 0 = Updates every frame, 1 = Updates every second frame, etc.

cl_render_ssao <value>

Type: String

1 = Screen space ambient occlusion enabled, 0 = disabled.

cl_render_texture_filtering <value>

Type: String

0 = Nearest Neighbour, 1 = Bilinear Filtering, 2 = Trilinear Filtering, 3 = Antisotropic Filtering x2, 4 = Antisotropic Filtering x4, 5 = Antisotropic Filtering x8, 6 = Antisotropic Filtering x16

cl_render_texture_quality <value>

Type: String

0 = Minimal, 1 = Low, 2 = Medium, 3 = High, 4 = Very High.

cl_render_tone_mapping <value>

Type: String

Changes the tone-mapping algorithm to use. -1 = Use the default, 0 = Gamma correction only, 1 = Reinhard, 2 = Hejil-Richard, 3 = Uncharted, 4 = Aces, 5 = Gran Turismo

cl_render_vr_enabled <value>

Type: String

Turns support for virtual reality on or off.

cl_render_vsync_enabled <value>

Type: String

Turns vertical sync on or off.

cl_render_window_mode <value>

Type: String

0 = Fullscreen, 1 = Windowed, 2 = Windowed no-border.

cl_show_fps <value>

Type: String

Displays the current fps at the top left of the screen if enabled.

cl_steam_audio_ambisonics_order <value>

Type: String

The amount of directional detail in the simulation results. Range: [0,3].

cl_steam_audio_enabled <value>

Type: String

Enables or disables steam audio.

cl_steam_audio_ir_duration <value>

Type: String

The time delay between a sound being emitted and the last audible reflection. Range: [0.5,4.0].

cl_steam_audio_number_of_bounces <value>

Type: String

The maximum number of times any ray can bounce within the scene. Range: [1,32].

cl_steam_audio_number_of_diffuse_samples <value>

Type: String

The number of directions to consider when a ray bounces off a diffuse (or partly diffuse) surface. Range: [32,4096].

cl_steam_audio_number_of_rays <value>

Type: String

The number of rays to trace from the listener. Range: [1024,131072].

cl_steam_audio_propagation_delay_enabled <value>

Type: String

Enables or disables propagation delay if steam audio is enabled.

cl_steam_audio_reverb_enabled <value>

Type: String

Enables the reverb DSP effect.

cl_steam_audio_reverb_indirect_binaural <value>

Type: String

Spatialize reverb using HRTF.

cl_steam_audio_reverb_simulation_type <value>

Type: String

Real-time or baked. 0 = Real-time, 1 = Baked.

cl_steam_audio_spatialize_air_absorption <value>

Type: String

Enable air absorption.

cl_steam_audio_spatialize_direct_binaural <value>

Type: String

Spatialize direct sound using HRTF.

cl_steam_audio_spatialize_direct_level <value>

Type: String

Relative level of direct sound. Range: [0,1].

cl_steam_audio_spatialize_distance_attenuation <value>

Type: String

Enable distance attenuation. If disabled, regular distance attenuation will be used instead.

cl_steam_audio_spatialize_enabled <value>

Type: String

Enables the spatialize DSP effect.

cl_steam_audio_spatialize_hrtf_interpolation <value>

Type: String

HRTF interpolation. 0 = Nearest, 1 = Bilinear.

cl_steam_audio_spatialize_indirect <value>

Type: String

Enable indirect sound.

cl_steam_audio_spatialize_indirect_binaural <value>

Type: String

Spatialize indirect sound using HRTF.

cl_steam_audio_spatialize_indirect_level <value>

Type: String

Relative level of indirect sound. Range: [0,10].

cl_steam_audio_spatialize_occlusion_method <value>

Type: String

Direct occlusion algorithm. 0 = Raycast, 1 = Partial.

cl_steam_audio_spatialize_occlusion_mode <value>

Type: String

Direct occlusion and transmission mode. 0 = Off, 1 = On, No Transmission, 2 = On, Frequency Independent Transmission, 3 = On, Frequency Dependent Transmission.

cl_steam_audio_spatialize_simulation_type <value>

Type: String

Real-time or baked. 0 = Real-time, 1 = Baked

cl_steam_audio_spatialize_static_listener <value>

Type: String

Uses static listener.

cl_updaterate <value>

Type: String

The amount of times per second user input is being transmitted to the server.

cl_voice_volume <value>

Type: String

Volume scale for voice (Coming from players and NPCs).

cl_vr_hmd_view_enabled <value>

Type: String

Enables or disables the view for virtual reality (Only works if virtual reality has been enabled in the menu.).

cl_vr_mirror_window_enabled <value>

Type: String

Enables or disables the mirror window for virtual reality (Only works if virtual reality has been enabled in the menu.).

cl_water_surface_simulation_edge_iteration_count <value>

Type: String

The more iterations, the more detailed the water simulation will be, but at a great performance cost.

cl_water_surface_simulation_enable_gpu_acceleration <value>

Type: String

If enabled, water surfaces will be simulated on the GPU instead of the CPU. This should greatly improve performance in most cases.

cl_water_surface_simulation_spacing <value>

Type: String

Grid spacing for simulating water surfaces. Lower values will result in a more detailed simulation, but at a great performance cost.

cl_window_resolution <value>

Type: String

Changes the window resolution.

debug_ai_navigation <value>

Type: String

Displays the current navigation path for all NPCs.

debug_bsp_lock <value>

Type: String

Locks BSP culling so it doesn't change when the camera moves.

debug_bsp_nodes <value>

Type: String

Prints and draws various debug information about the binary space partitioning of the level (if available). 1 = Draw leaf of camera location and all visible leaves, 2 = Same, but doesn't update continuously.

debug_hide_gui 1/0

Type: Boolean

Disables GUI rendering.

debug_nav_show_meshes <value>

Type: String

Shows the navigation mesh for this map.

debug_occlusion_culling_freeze_camera <value>

Type: String

Freezes the current camera position in place for occlusion culling.

debug_particle_blob_show_neighbor_links <value>

Type: String

Displays the links to adjacent neighbors for blob particles.

debug_physics_draw <value>

Type: String

Draws various physics information as wireframe.

debug_profiling_enabled <value>

Type: String

Enables profiling timers.

debug_render_lock_render_command_buffers <value>

Type: String

Locks the current render command buffers and causes them to not be re-created every frame.

debug_render_lock_render_queues <value>

Type: String

Locks the current render queues and effectively disables the render queue thread.

debug_render_normals <value>

Type: String

Displays all normals as lines in the world. 0 = Disabled, 1 = Display normals, 2 = Display normals, tangents and bitangents, 3 = Display face normals, 4 = Display tangents, 5 = Display bitangents.

debug_render_octree_dynamic_draw <value>

Type: String

Draws the octree for dynamic objects.

debug_render_octree_static_draw <value>

Type: String

Draws the octree for static world geometry.

debug_steam_audio_probe_boxes <value>

Type: String

Displays the sound probe spheres for the current map.

host_timescale <value>

Type: String

Specifies the timescale. A value of 2 means the game will run at twice the default speed.

log_enabled <value>

Type: String

0 = Log disabled; 1 = Log errors only; 2 = Log errors and warnings; 3 = Log all console output

log_file <value>

Type: String

The log-file the console output will be logged to.

lua_open_editor_on_error <value>

Type: String

1 = Whenever there's a Lua error, the engine will attempt to automatically open a Lua IDE and open the file and line which caused the error.

net_graph <value>

Type: String

Displays a graph about current network transmissions.

net_library <netLibrary>

Type: String

The underlying networking library to use for multiplayer games.

password <value>

Type: String

Password which will be used for the next connection attempt.

phys_engine <physEngie>

Type: String

The underlying physics engine to use.

playername <value>

Type: String

Local player name.

rcon_password <value>

Type: String

Specifies a password which can be used to run console commands remotely on a server. If no password is specified, this feature is disabled.

render_api <renderApi>

Type: String

The underlying rendering API to use.

render_bloom_amount <value>

Type: String

Number of render passes to use for bloom.

render_bloom_enabled <value>

Type: String

Enables or disables bloom and glow effects.

render_bloom_resolution <value>

Type: String

The width for the bloom texture. The height will be calculated using the aspect ratio of the renderer.

render_clear_scene <value>

Type: String

0 = Default, 1 = Screen is cleared before scene is rendered.

render_clear_scene_color <value>

Type: String

Clear color to use if render_clear_scene is set to 1.

render_csm_max_distance <value>

Type: String

The maximum distance at which csm shadows are casted. A larger distance means lower quality.

render_debug_mode 1/0

Type: Boolean

0 = Disabled, 1 = Ambient Occlusion, 2 = Albedo Colors, 3 = Metalness, 4 = Roughness, 5 = Diffuse Lighting, 6 = Normals, 7 = Normal Map, 8 = Reflectance, 9 = IBL Prefilter, 10 = IBL Irradiance, 11 = Emission, 12 = Lightmaps, 13 = Lightmap Uvs, 14 = Unlit, 15 = Show CSM cascades, 16 = Shadow Map Depth, 17 = Forward+ Heatmap.

render_draw_dynamic <value>

Type: String

1 = Default, 0 = Don't draw props.

render_draw_glow <value>

Type: String

1 = Default, 0 = Glow-effects aren't drawn.

render_draw_particles <value>

Type: String

1 = Default, 0 = Particles aren't drawn.

render_draw_scene <value>

Type: String

1 = Default, 0 = Scene isn't drawn.

render_draw_sky <value>

Type: String

1 = Default, 0 = Sky isn't drawn.

render_draw_static <value>

Type: String

1 = Default, 0 = Don't draw level geometry.

render_draw_translucent <value>

Type: String

1 = Default, 0 = Don't draw translucent objects.

render_draw_view <value>

Type: String

1 = Default, 0 = View-Models aren't drawn.

render_draw_water <value>

Type: String

1 = Default, 0 = Water isn't drawn.

render_draw_world <value>

Type: String

1 = Default, 2 = Wireframe, 0 = Don't draw world.

render_dynamic_lighting_enabled 1/0

Type: Boolean

Enables or disables dynamic lighting.

render_dynamic_shadows_enabled 1/0

Type: Boolean

Enables or disables dynamic shadows.

render_forwardplus_tile_size <value>

Type: String

Changes the tile size used for the light culling in forward+.

render_hdr_max_exposure <value>

Type: String

Expore will be clamped at this value. Can be overwritten by the map.

render_ibl_enabled 1/0

Type: Boolean

Enables or disables image-based lighting.

render_instancing_enabled 1/0

Type: Boolean

Enables or disables instanced rendering.

render_instancing_threshold [2,inf]

Type: UInt32

The threshold at which to start instancing entities if instanced rendering is enabled (render_instancing_threshold). Must not be lower than 2!

render_multithreaded_rendering_enabled 1/0

Type: Boolean

Enables or disables multi-threaded rendering. Some renderers (like OpenGL) don't support multi-threaded rendering and will ignore this flag.

render_normalmapping_enabled <value>

Type: String

1 = Normal mapping enabled, 0 = disabled.

render_pp_nightvision <value>

Type: String

Enables or disables night vision.

render_pp_nightvision_color_amplification <value>

Type: String

Amplification factor for pixels with low luminance.

render_pp_nightvision_luminance_threshold <value>

Type: String

Changes the luminance threshold for the night vision post-processing effect.

render_pp_nightvision_noise_texture <value>

Type: String

Noise texture to use for the night vision post-processing effect.

render_queue_entities_per_worker_job [1,50]

Type: UInt32

Number of entities for each job processed by a worker thread.

render_queue_worker_jobs_per_batch [0,10]

Type: UInt32

Number of worker jobs to accumulate in a batch before assigning a worker.

render_queue_worker_thread_count [1,10]

Type: UInt32

Number of threads to use for generating render queues.

render_shadow_quality <value>

Type: String

Shadowmap Quality. This affects the detail of the object shadows that are being rendered

render_technique <value>

Type: String

0 = Rasterization, 1 = Raytracing

render_texture_filtering <value>

Type: String

0 = Nearest-Neighbor, 1 = Bilinear, 2 = trilinear.

render_vsync_enabled <value>

Type: String

1 = Vertical sync enabled, 0 = disabled.

sh_lua_remote_debugging <value>

Type: String

0 = Remote debugging is disabled; 1 = Remote debugging is enabled serverside; 2 = Remote debugging is enabled clientside. Cannot be changed during an active game. Also requires the "-luaext" launch parameter. Remote debugging cannot be enabled clientside and serverside at the same time.

sh_mount_external_game_resources <value>

Type: String

If set to 1, the game will attempt to load missing resources from external games.

sky_override <value>

Type: String

Replaces the material for all current skyboxes with the specified material.

steam_steamworks_enabled <value>

Type: String

Enables or disables steamworks.

sv_acceleration <value>

Type: String

Player acceleration. If this is too low, the player will be unable to reach full movement speed due to friction forces.

sv_allowcslua <value>

Type: String

1 = Clients can run own clientside scripts, 0 = Custom clientside scripts aren't allowed.

sv_allowdownload <value>

Type: String

Specifies whether clients are allowed to download resources from the server.

sv_allowupload <value>

Type: String

Specifies whether clients are allowed to upload resources to the server (e.g. spraylogos).

sv_cheats <value>

Type: String

Allows clients to run console commands which are marked as cheats if set to 1.

sv_debug_physics_draw <value>

Type: String

Draws various serverside physics information as wireframe.

sv_friction <value>

Type: String

Global friction scale.

sv_gamemode <value>

Type: String

Gamemode which should be used when starting a game.

sv_gravity <value>

Type: String

Specifies the default gravity force and direction. Certain entities may overwrite this for themselves.

sv_maxplayers <value>

Type: String

Specifies the maximum amount of players that are allowed to join the server.

sv_noclip_speed <value>

Type: String

The speed at which players can move in noclip mode.

sv_password <value>

Type: String

Sets a password for the server. No player will be able to join, unless they provide the correct password using the 'password' command.

sv_physics_simulation_enabled <value>

Type: String

Enables or disables physics simulation.

sv_player_air_move_scale <value>

Type: String

The scale at which players are still able to move while in the air. 0 = Players can't move until they've landed, 1 = Players can move freely while in the air

sv_port_tcp <value>

Type: String

TCP port which will be used when starting a server.

sv_port_udp <value>

Type: String

UDP port which will be used when starting a server.

sv_require_authentication 1/0

Type: Boolean

If enabled, clients will have to authenticate via steam to join the server.

sv_restitution <value>

Type: String

Global restitution scale.

sv_servername <value>

Type: String

The server name which will be displayed in the server browser.

sv_tickrate <value>

Type: String

Specifies the tickrate. A higher tickrate means smoother and more reliable physics, but also more data to transmit to clients. Higher values can result in more lag for clients.

sv_timeout_duration <value>

Type: String

Amount of time until a client is forcibly dropped if no data has been received.

sv_use_p2p_if_available <value>

Type: String

Use a peer-to-peer connection if the selected networking layer supports it.

sv_water_surface_simulation_edge_iteration_count <value>

Type: String

The more iterations, the more detailed the water simulation will be, but at a great performance cost.

sv_water_surface_simulation_shared <value>

Type: String

If enabled, water surface simulation will be shared between client and server (Simulation is only performed once). This will only have an effect in single-player or on listen servers.

Commands

+attack

Triggers the primary fire for the player character's equipped weapon.

+attack2

Triggers the secondary fire for the player character's equipped weapon.

+attack3

Triggers the tertiary fire for the player character's equipped weapon.

+attack4

Triggers the quaternary fire for the player character's equipped weapon.

+backward

Moves the player character backward while active.

+crouch

Prompts the player character to crouch.

+forward

Moves the player character forward while active.

+jump

Prompts the player character to jump.

+left

Moves the player character left while active.

+reload

Triggers the reload for the player character's equipped weapon.

+right

Moves the player character right while active.

+sprint

Enables sprinting while moving forward.

+turn_down

Continuously turns the player character leftwards.

+turn_left

Continuously turns the player character leftwards.

+turn_right

Continuously turns the player character leftwards.

+turn_up

Continuously turns the player character leftwards.

+use

Prompts the player character to use whatever is in front of him.

+walk

Enables walking while moving forward.

-attack

Triggers the primary fire for the player character's equipped weapon.

-attack2

Triggers the secondary fire for the player character's equipped weapon.

-attack3

Triggers the tertiary fire for the player character's equipped weapon.

-attack4

Triggers the quaternary fire for the player character's equipped weapon.

-backward

Moves the player character backward while active.

-crouch

Prompts the player character to crouch.

-forward

Moves the player character forward while active.

-jump

Prompts the player character to jump.

-left

Moves the player character left while active.

-reload

Triggers the reload for the player character's equipped weapon.

-right

Moves the player character right while active.

-sprint

Enables sprinting while moving forward.

-turn_down

Continuously turns the player character leftwards.

-turn_left

Continuously turns the player character leftwards.

-turn_right

Continuously turns the player character leftwards.

-turn_up

Continuously turns the player character leftwards.

-use

Prompts the player character to use whatever is in front of him.

-walk

Enables walking while moving forward.

asset_clear_unused

Clears all unused assets from memory.

asset_clear_unused_materials

Clears all unused materials from memory.

asset_clear_unused_models

Clears all unused models from memory.

asset_clear_unused_textures

Clears all unused textures from memory.

bind

Binds a key to a command string. Usage: bind <key> <command>

bind_keys

Prints a list of all bindable keys to the console.

cl_debug_netmessages

Prints out debug information about recent net-messages.

cl_entities

Prints a list of all current clientside entities in the world.

cl_find

Finds similar console commands to whatever was given as argument.

cl_gpu_timer_queries_dump

Prints all timer query results to the console.

cl_list

Prints a list of all clientside console commands to the console.

cl_send

Sends a text message to the server and displays it in the console. Usage: cl_send <message>

cl_send_udp

Sends a text message to the server via UDP and displays it in the console. Usage: cl_send_udp <message>

cl_steam_audio_reload_scene

Reloads the steam audio scene cache.

clear

Clears everything in the console.

clear_cache

Clears all of the cached engine files.

closeserver

Closes the server (if active) and drops all connected clients.

connect

Attempts to connect to the specified server. Usage: connect <ipV6/ipV4> or connect <steamId> or without arguments to re-try the last attempt.

crash

Forces the engine to crash.

credits

Prints a list of developers.

debug_ai_schedule

Prints the current schedule behavior tree for the specified NPC on screen.

debug_ai_schedule_print

Prints the current schedule behavior tree for the specified NPC.

debug_aim_info

Prints information about whatever the local player is looking at.

debug_audio_aux_effect

Applies a global DSP effect. Usage: debug_audio_aux_effect <dspName> <gain>

debug_audio_sounds

Prints information about all active server- and clientside sounds to the console.

debug_dump_shader_code

Dumps the glsl code for the specified shader.

debug_font

Displays the glyph map for the specified font. If no arguments are specified, all available fonts will be listed. Usage: debug_font <fontName>

debug_font_glyph_map

Displays the glyph map for the specified font.

debug_glow_bloom

Displays the scene glow texture on screen. Usage: debug_glow_bloom <1/0>

debug_gui_cursor

Prints information about the GUI element currently hovered over by the cursor.

debug_hdr_bloom

Displays the scene bloom texture on screen. Usage: debug_hdr_bloom <1/0>

debug_hitboxes

Displays the hitboxes for the entity under the crosshair, or the entity with the given name (If it was specified.).

debug_light_shadowmap

Displays the depth map for the given light on screen. Call without arguments to turn the display off. Usage: debug_light_shadowmap <lightEntityIndex>

debug_light_sources

Prints debug information about all light sources in the scene.

debug_lightmaps

Displays the lightmaps for the loaded level.

debug_nav_path_end

Sets an end point for a navigation path. Use debug_nav_path_start to set the start point. The path will be drawn in the scene once both points have been set.

debug_nav_path_start

Sets a start point for a navigation path. Use debug_nav_path_end to set the end point.

debug_particle_alpha_mode

Specifies the blend mode arguments for particle systems that use the 'custom' alpha mode. Argument order: <srcColorBlendFactor> <dstColorBlendFactor> <srcAlphaBlendFactor> <dstAlphaBlendFactor> <opColor> <opAlpha>. Blend factor options: zero, one, src_color, one_minus_src_color, dst_color, one_minus_dst_color, src_alpha, one_minus_src_alpha, dst_alpha, one_minus_dst_alpha, constant_color, one_minus_constant_color, constant_alpha, one_minus_constant_alpha, src_alpha_saturate, src1_color, one_minus_src1_color, src1_alpha, one_minus_src1_alpha Operation options: add, subtract, reverse_subtract, min, max

debug_pbr_ibl

Displays the irradiance, prefilter and brdf map for the closest cubemap.

debug_prepass

Displays prepass buffers to screen.

debug_profiling_physics_end

Prints physics profiling information for the last simulation step.

debug_profiling_physics_start

Prints physics profiling information for the last simulation step.

debug_profiling_print

Prints the last profiled times.

debug_render_depth_buffer

Draws the scene depth buffer to screen.

debug_render_info

Prints some timing information to the console.

debug_render_memory_budget

Prints information about the current GPU memory budget.

debug_render_memory_stats

Prints statistics about the current GPU memory usage.

debug_render_octree_dynamic_find

Finds the specified entity in the octree for dynamic objects.

debug_render_octree_dynamic_print

Prints the octree for dynamic objects to the console, or a file if a file name is specified.

debug_render_octree_static_print

Prints the octree for static world geometry to the console, or a file if a file name is specified.

debug_render_scene

Displays scene to screen.

debug_render_stats

Prints information about the next frame.

debug_render_validation_error_enabled

Enables or disables the specified validation error.

debug_ssao

Displays the ssao buffers to screen.

debug_steam_audio_dump_scene

Saves the steam audio scene as OBJ file.

debug_texture_mipmaps

Displays the mipmaps of the given texture. Usage: debug_texture_mipmaps <texturePath>

debug_textures

Prints information about the currently loaded textures.

debug_water

Displays the reflection, refraction and refraction depth map for the given water-entity on screen. Call without arguments to turn the display off. Usage: debug_light_water <waterEntityIndex>

disconnect

Disconnects from the server (if a connection is active), or closes the game if in single player mode.

drop

Drops the player's active weapon.

echo

Prints something to the console. Usage: echo <message>

ent_create

Creates and spawns a new entity with the given class name at the position in the world the player is looking at. Usage: ent_create <className>

ent_input

Triggers the given input on the specified entity. Usage: ent_input <entityName/entityClass> <input>

ent_remove

Removes the specified entity, or the entity the player is looking at if no argument is provided. Usage: ent_remove <entityName/className>

ent_scale

Changes the scale of the specified entity. Usage: ent_input <entityName/entityClass> <scale>

entities

Prints a list of all current serverside entities in the world.

exec

Executes a config file. Usage exec <fileName>

exit

Exits the game.

find

Finds similar console commands to whatever was given as argument.

flashlight

Toggles the flashlight.

fps

Prints the current fps to the console.

getang

Prints the current angles of the local player to the console.

getcamang

Prints the current angles of the camera to the console.

getcampos

Prints the current position of the camera.

getpos

Prints the current position of the local player to the console.

give_ammo

Gives the local player the specified ammunition.

give_weapon

Gives the local player the specified weapon and deploys it.

godmode

Toggles god mode. Requires cheats to be enabled (Or single player mode running).

heartbeat

Instantly sends a heartbeat to the master server.

help

Find help about a convar/concommand.

hurtme

Applies the specified amount of damage to self.

keymappings

Prints a list of all active key bindings to the console.

kick

Kicks the specified player for the given reason. Usage: kick <playerId/playerName> <reason>

list

Prints a list of all serverside console commands to the console.

list_maps

Prints a list of all available list to the console.

listmaps

load

loc_reload

Reloads all localization files.

log

Adds the specified message to the engine log. Usage: log <msg>.

lua_compile

Opens the specified lua-file and outputs a precompiled file with the same name (And the extension '.clua').

lua_exec

Opens and executes a lua-file on the server.

lua_exec_cl

Opens and executes a lua-file on the client.

lua_help

Prints information about the specified function, library or enum (or the closest candiate). Usage: lua_help <function/library/enum>.

lua_reload_entities

Reloads the scripts for all registered lua entity classes.

lua_reload_entity

Reloads the scripts for the given entity class. Usage: lua_reload_entity <className>

lua_reload_weapon

Reloads the scripts for the given weapon class. Usage: lua_reload_weapon <className>

lua_reload_weapons

Reloads the scripts for all registered lua weapon classes.

lua_run

Runs a lua command on the server lua state.

lua_run_cl

Runs a lua command on the client lua state.

lua_run_gui

Runs a lua command on the GUI lua state.

map

Loads the given map immediately. Usage: map <mapName>

map_build_reflection_probes

Build all reflection probes in the map. Use the '-rebuild' argument to clear all current IBL textures first. Use 'debug_pbr_ibl' to check the probes after they have been built.

map_rebuild_lightmaps

Rebuilds the lightmaps for the current map. Note that this will only work if the map was compiled with lightmap uvs.

nav_generate

Generates a navigation mesh for the current map and saves it as a navigation file.

nav_reload

Reloads the navigation mesh for the current map.

next_weapon

Switches to the next weapon in the player's inventory.

noclip

Toggles noclip mode. Requires cheats to be enabled (Or single player mode running).

notarget

Toggles notarget mode. Requires cheats to be enabled (Or single player mode running).

previous_weapon

Switches to the previous weapon in the player's inventory.

quit

Exits the game.

rcon

Sends the given command to the server and runs it serverside. Usage: rcon <command>

reloadmaterial

Reloads the given material (and textures associated with it). Usage: reloadmaterial <materialFile>

reloadmaterials

Reloads all loaded materials and textures.

render_api_info

Prints information about the current render API to the console.

save

screenshot

Writes the contents of the screen into a .tga-file in 'screenshots/'. The name of the file will start with the map's name, or 'pragma' if no map is loaded. After the name follows an ID which is incremented each time. You can use 'screenshot 1' to take a screenshot without the GUI (Ingame

only).

setang

Changes the angles of the local player. Requires cheats to be enabled. Usage: setang <pitch> <yaw> <roll>

setpos

Changes the position of the local player. Requires cheats to be enabled. Usage: setpos <x> <y> <z>

shader_list

Prints a list of all currently loaded shaders

shader_optimize

Uses LunarGLASS to optimize the specified shader.

shader_reload

Reloads a specific shader, or all registered shaders if no arguments are given. Usage: shader_reload <shaderName>

sound_play

Precaches and immediately plays the given sound file. Usage: sound_play <soundFile>

sound_stop

Stops all sounds that are currently playing.

startserver

Starts an internet server. Requires a running game.

status

Prints information about the current connection to the console.

strip_weapons

Strips all weapons from the local player.

suicide

Commit suicide.

sv_debug_netmessages

Prints out debug information about recent net-messages.

sv_send

Sends a text message to all connected clients and displays it in the console. Usage: sv_send <message>

sv_send_udp

Sends a text message to all connected clients via UDP and displays it in the console. Usage: sv_send_udp <message>

thirdperson

Toggles between first- and third-person mode.

toggle

Toggles the specified console variable between 0 and 1.

toggleconsole

Toggles the developer console.

udm_convert

Converts a UDM file from binary to ASCII or the other way around.

udm_validate

Validates the specified UDM file.

unbind

Unbinds the given key.

unbindall

Unbinds all keys.

version

Prints the current engine version to the console.

vk_dump_extensions

Dumps all available vulkan extensions to 'vk_extensions.txt'.

vk_dump_features

Dumps all vulkan features to 'vk_features.txt'.

vk_dump_format_properties

Dumps all vulkan format properties to 'vk_format_properties.txt'.

vk_dump_image_format_properties

Dumps all vulkan image format properties to 'vk_image_format_properties.txt'.

vk_dump_layers

Dumps all available vulkan layers to 'vk_layers.txt'.

vk_dump_limits

Dumps all vulkan limitations to 'vk_limits.txt'.

vk_dump_memory_stats

Dumps information about currently allocated GPU memory.

vk_print_memory_stats

Prints information about currently allocated GPU memory to the console.

Game Mounting

Pragma can automatically import assets that can't be found in its native formats from one of the mounted games. The default list of mounted games includes all Valve Source Engine games, as well as a few popular mods, but in some cases you may want to add custom entries. This is currently only possible by editing the list of mounted games in a text-editor manually.

To do so, go to "Pragma/cfg" and open "mounted_games.udm" in a text-editor. Scroll to the end of the file and add the following entry:

```
"<identifier>"
{
	$bool enabled 1
	"steam"
	{
		$uint32 app_id <appId>
		$string game_path "<gamePath>"
		$bool mount_workshop <mountWs>
	}
	"engine"
	{
		$string name "source_engine"
	}
}
```

Replace the following attributes:

- `<identifier>:` A unique identifier for this game. Make sure the name is lowercase, has no spaces and no special characters, and isn't already used for any other entries.
- `<appId>:` The Steam App Id of the game. You can find it by entering the game's name on SteamDB.
- `<gamePath>:` The path to the game's assets, relative to "steamapps/". This path should contain a "models", "materials" and "maps" directory. For example, if the full path is "E:/SteamLibrary/steamapps/common/Counter-Strike Global Offensive/csgo", the `<gamePath>` should be "common/Counter-Strike Global Offensive/csgo/". Make sure to use this exact format (forward slashes; not starting with a slash; ending with a slash).
- `<mountWs>:` Set to `1` if you want to mount the workshop for the specified game, otherwise `0`.

You can use the other entries as a reference. If you've made a syntax error, it should appear in the console the next time you start Pragma. You can also run the console command `udm_validate cfg/mounted_games.udm` to ensure the file is syntactically correct.

If your new entry doesn't seem to have any effect, you can start Pragma with the `-verbose -console terminal` launch parameters, and it should print out a bunch of information about the mounted games and any errors.

Mounting custom SFM user folders

If your assets are located in a custom directory under "steamapps/common/SourceFilmmaker/game/", it will not be mounted by default. To mount it, open "Pragma/cfg/mounted_games.udm" in a text-editor, find the "sfm" entry and add the path to the "game_paths" list (replace `<your_custom_folder>` with the name of your custom directory):

```
"sfm"
{
	$bool enabled 1
	$string localization_name "mount_game_sfm"
	$int32 priority 950
	"steam"
	{
		$uint32 app_id 1840
		$array game_paths [string][
			"common/SourceFilmmaker/game/usermod/",
			"common/SourceFilmmaker/game/workshop/",
			"common/SourceFilmmaker/game/tf_movies/",
			"common/SourceFilmmaker/game/tf/",
			"common/SourceFilmmaker/game/hl2/",
			"common/SourceFilmmaker/game/left4dead2_movies/",
			"common/SourceFilmmaker/game/platform/",
			"common/SourceFilmmaker/game/portal2/",
			"common/SourceFilmmaker/game/<your_custom_folder>/"
		]
		$bool mount_workshop 1
	}
	"engine"
	{
		$string name "source_engine"
```

```
}
}
```

Developer Documentation

Building Pragma

You can also find these build instructions on the Pragma repository on GitHub.

Build Requirements

- ~50 GiB of disk space
- CMake 3.21.4 or newer
- Python 3.9.5 or newer

Windows

- Visual Studio 2022 or newer

Linux

- clang-14 or newer (Pragma is *not* compatible with gcc!)

Build Instructions

Launch a command-line interface and clone the pragma repository (with submodules) into a directory of your choice:

```
git clone https://github.com/Silverlan/pragma.git --recurse-submodules
```

After that you can simply navigate to the pragma directory and run the python build-script, which will automatically download all dependencies, configure CMake, and build and install the project (this will take several hours):

```
python build_scripts/build.py --with-pfm --with-all-pfm-modules --with-vr
```

> If you don't need the filmmaker, you can omit the `--with-pfm --with-all-pfm-modules` arguments, which will significantly reduce the build time and the required amount of disk space.

Linux

> Before running the build script, you will have to install the following packages:

```
# Required for the build script
sudo apt-get install python3

# Required for Pragma core
sudo apt install build-essential
sudo add-apt-repository ppa:savoury1/llvm-defaults-14
sudo apt update
sudo apt install clang-14
sudo apt install libstdc++-12-dev
sudo apt install libstdc++6
sudo apt-get install patchelf

# Required for Vulkan
sudo apt-get -qq install -y libwayland-dev libxrandr-dev

sudo apt-get install libxcb-keysyms1-dev
sudo apt-get install xcb libxcb-xkb-dev x11-xkb-utils libx11-xcb-dev libxkbcommon-x11-dev

# Required for GLFW
sudo apt install xorg-dev

# Required for OIDN
sudo apt install git-lfs

# Required for Cycles
sudo apt-get install subversion

# Required for Curl
sudo apt-get install libssl-dev
sudo apt install libssh2-1

# Required for OIIO
sudo apt-get install python3-distutils
```

Once the build script has been completed, you should find the build files in `pragma/build`, and the install files in `pragma/build/install`. The `install` directory should contain everything you need to run Pragma.

If you make any code changes to the core engine code, you can build the `pragma-install` target to build them. This will also re-install the binaries.

If you make any code changes to a module, you will have to build the module build target first, and then build `pragma-install` afterwards.

Build Customization

Running the build-script with the arguments above will build and install Pragma and the Pragma Filmmaker with all dependencies. Alternatively you can also configure the build to your liking with the following parameters:

Parameter	Description	Default
`--help`	Display this help	
`--generator <generator>`	The generator to use.	`Visual Studio 17 2022` (On Windows) `Unix Makefiles` (On Linux)
`--c-compiler`	[Linux only] The C-compiler to use.	`clang-14`
`--cxx-compiler`	[Linux only] The C++-compiler to use.	`clang++-14`
`--with-essential-client-modules <1/0>`	Include essential modules required to run Pragma.	1
`--with-common-modules <1/0>`	Include non-essential but commonly used modules (e.g. audio and physics modules).	1
`--with-pfm <1/0>`	Include the Pragma Filmmaker.	0

`--with-core-pfm-modules <1/0>`	Include essential PFM modules.	1
`--with-all-pfm-modules <1/0>`	Include non-essential PFM modules (e.g. chromium and cycles).	0
`--with-vr <1/0>`	Include Virtual Reality support.	0
`--build <1/0>`	Build Pragma after configurating and generating build files.	1
`--build-config <config>`	The build configuration to use.	`RelWithDebInfo`
`--build-directory <path>`	Directory to write the build files to. Can be relative or absolute.	`build`
`--deps-directory <path>`	Directory to write the dependency files to. Can be relative or absolute.	`deps`
`--install-directory <path>`	Installation directory. Can be relative (to build directory) or absolute.	`install`
`--verbose <1/0>`	Print additional debug information.	0
`--module <moduleName>:<gitUrl>`	Custom modules to install. Use this argument multiple times to use multiple modules.	

Example for using the `--module` parameter:

```
--module pr_physx:"https://github.com/Silverlan/pr_physx.git"
```

If you want to create your own binary module, please check out the article on binary modules.

Binary Modules

Binary modules allow you to change or extend the behavior of Pragma without having to change the core source code. Binary modules are written in C++ and loaded during runtime using Lua (with some special exceptions). Some example modules are:

- pr_chromium: Adds an integrated Chromium-based Web Browser to Pragma
- pr_openvr: Adds virtual reality support to Pragma
- pr_bullet: Adds support for the Bullet physics engine to Pragma
- pr_physx: Adds support for the PhysX physics engine to Pragma
- pr_curl: Adds support for the curl library to Pragma
- pr_sqlite: Adds SQLite support to Pragma

Installing Modules

If the binary module is available on GitHub and has at least one release, you can use the following console command in Pragma to install the module directly:

```
install_module <GitHubUser>/<RepoName>
```

For instance, if you want to install the pr_curl module, simply run `install_module Silverlan/pr_curl` in the console, which will automatically download the module and install it.

Alternatively, if a binary module comes with prebuilt binaries, you can simply extract the archive to your Pragma installation directory, the file structure in the archive should ensure that its placed in the correct location. Simple modules that consist of a single library file can usually be found in the "modules" directory, while more complex modules (like chromium) reside in their own sub-directory within "modules".

The module can then be loaded with a simple Lua-script:

```
local moduleName = "chromium/pr_chromium"
local result = engine.load_library(moduleName)
```

```
if(result ~= true) then
    console.print_warning("Failed to load \"" .. moduleName .. "\" module: " .. result)
    return
end
```

The module name is the path to the binary file, without the "modules/" prefix and without the file extension. On Linux the module file also has the "lib" prefix, which has to be omitted as well. For instance, if the module filename is "modules/curl/libpr_curl.so", then the module name should be "curl/pr_curl".

Building Modules

If you want to build a module manually, you'll have to first build Pragma. You can then add the module to the build instructions by using one of the following methods:

Method 1) (Recommended)

Open `pragma/build_scripts/user_modules.py` in a text-editor and follow the instructions to add your module.

After that, simply re-run the build script.

> Tip: You can use the --rerun option to make the build script re-use the options used in the previous build.

Method 2)

You can also add the following option when running the build script:

```
--module <name>:<gitUrl>
```

For instance, if you want to add the `pr_chromium` module to the build:

```
--module pr_chromium:https://github.com/Silverlan/pr_chromium
```

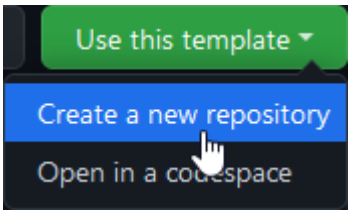
(You can use the `--module` argument multiple times if you want to add more than one module.)

Once the module has been added to the build, you can simply build the `pragma-install` target. It will build the module and install it automatically. You should then be able to find the module within the "modules" directory of the pragma installation folder.

Custom Modules

Setting up your own custom module is very simple and only takes a few minutes using the Pragma module template repository on GitHub, which also includes workflows for automated builds and releases.

To do so, go to the template repository and create a new repository from it:



You can choose whatever name you want for the repository, but for consistency it is recommended that it should have the same name as the module, which means:

- It should start with the "pr_" prefix
- It should be all lowercase
- It should only contain letters and underscores

Some examples are: "pr_chromium", "pr_bullet", "pr_audio_fmod", etc.

Next click "Create repository from template" to generate the repository.

The generated repository now contains a bunch of template files, which still need to be initialized. To do so, open the file "template_pragma_module.json" on GitHub and edit the values within:



- **name**: A pretty name, which will appear in the generated readme.
- **module_name**: The internal name of the module (i.e. the name of the CMake target and the binaries). This name should always start with the prefix `pr_` and always be lowercase. It should match the repository name if possible.
- **install_directory**: The directory where the module should be installed to, relative to the Pragma installation. Please choose this value carefully:
  - If you know your module is going to be a single binary with no additional files, it's recommended to leave this value at the default ("modules/").
  - If your module requires additional files, it's recommended to change this value to a sub-directory, e.g. "modules/<moduleName>/".
  - If your module requires Lua-scripts or asset files, it's recommended to change this value to an addon path, e.g. "addons/<addonName>/modules/<moduleName>/".
- **release_directory**: The directory that will be used for the GitHub releases. This value usually depends on the *install_directory*:
  - If the *install_directory* is the default ("modules/"), leave this value empty. In this case only the binary file will be added to the GitHub release.
  - If the *install_directory* is a sub-directory in "modules/", set this value to the same sub-directory (i.e. the same value as *install_directory*).
  - If the *install_directory* points to an addon, set this value to the addon path (e.g. "addons/<addonName>/").

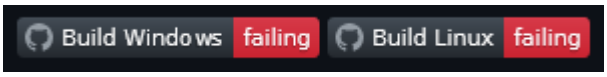
Example:

```
{
    "name": "Chromium",
    "module_name": "pr_chromium",
    "install_directory": "addons/chromium_browser/modules/chromium/",
    "release_directory": "addons/chromium_browser/"
}
```

When the module has been built, the binary will be located in "pragma/addons/chromium_browser/modules/chromium/pr_chromium.dll" (or "libpr_chromium.so" for Linux).

The GitHub release will contain all files that are within "pragma/addons/chromium_browser/". Users will be able to simply extract the release over their Pragma installation to install the module.

Once you have edited the values in the json-file, commit your changes. This will trigger a workflow which will initialize the repository with your values, which should take about two minutes. Simply refresh the page a few times, once you can see the following icons at the top of the readme, it means the initialization was completed:



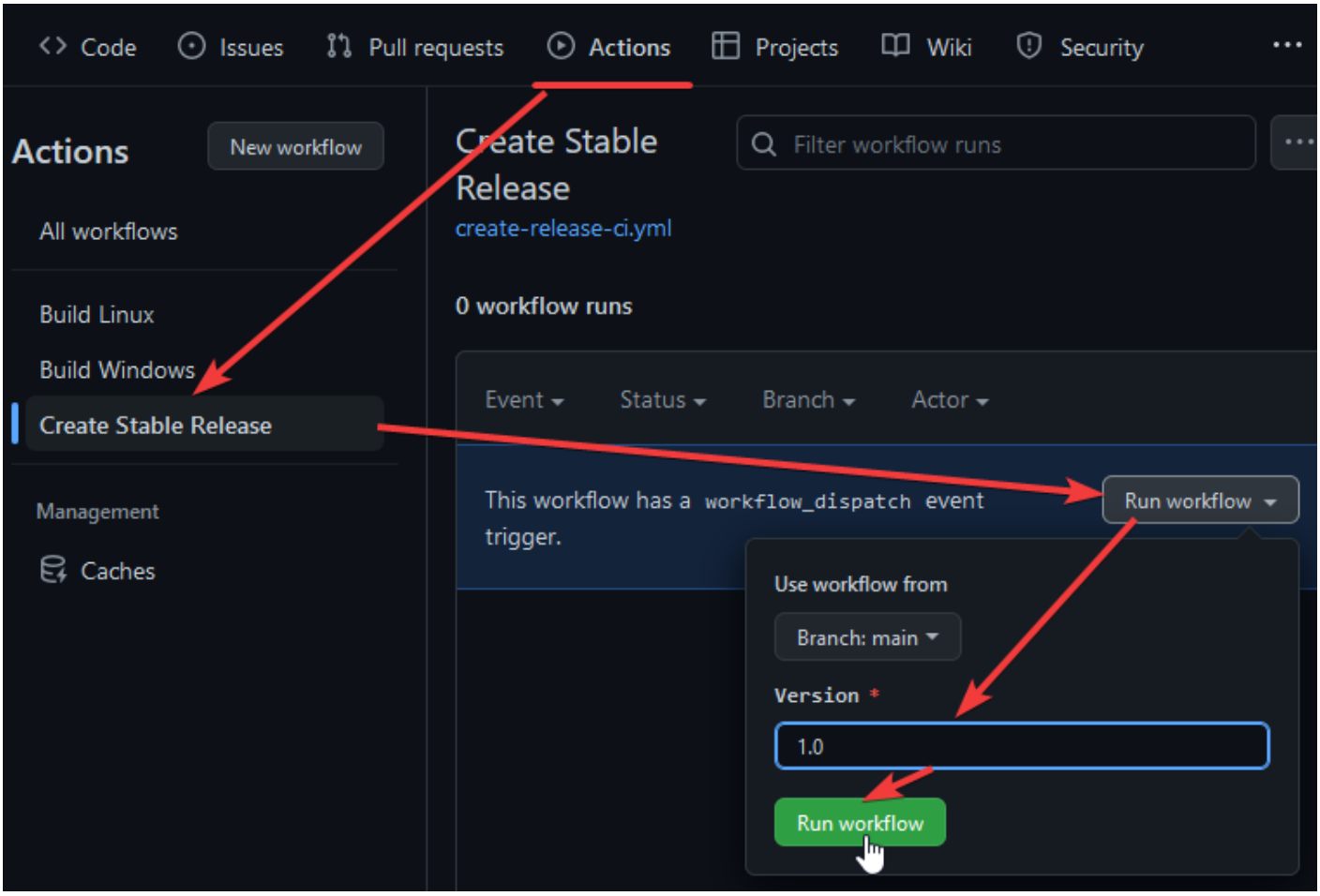
The initialization will also trigger the automated build workflows, which may take a few hours to run. Once they have completed, the two icons should turn green, and you should find the compiled binary files available for download for both Windows and Linux in the "Releases" section of the repository.

The build workflows will automatically trigger whenever you push any new commits to the repository and the release binaries will be updated every time. If there are any errors during the build process, the icons will say "failing" again, and you can check the workflow log for more information on what caused the failure.

Once you've set up the repository, you can follow the instructions in the section about building modules to add the module to the pragma build.

Stable Releases

If your module has reached a state that can be considered "stable", you can publish a stable release. To do so, wait for the regular build workflows to complete, then go to the "Actions" tab of your repository and select the "Create Stable Release" workflow, enter a version number and click "Run workflow":



This workflow should only take a minute, as it just publishes the latest binaries as a new release.

Testing

You can find a simple test Lua-script in the "examples" directory of your module. Follow the instructions in the commented section of the script to install and run it to test the module.

Advanced Building

If your module requires additional steps (such as downloading and building external dependencies) before it can be built, you can add those steps to the python script in "build_scripts/setup.py". This script will be executed automatically whenever the Pragma build script is run.

Advanced Installation

By default the module binary file will be installed to the "modules" directory of the Pragma installation. If you want to change this behavior, you can do so by editing the "CMakeInstall.txt" file.

Git Changelog

PFM (23-03-31)

Features

- "Base Properties" item in actor editor is now hidden for components that have no base properties (58ad221)
- actor bones now show in viewport when hovering over a body part (254a3c2)
- actor property values are now initialized from project after main component initialization and before entity spawn (e6d9b8d)
- add "OnChannelAdded" and "OnChannelRemoved" callbacks for animation manager (3028942)
- add "Save as Copy" menu bar option (14219d1)
- add additional tooltip information to actor editor constraint icons (2e721f8)
- add additional version information (c04a956)
- add AnimationManager::FindAnimation for finding/creating the animation of a specific actor (b500690)
- add axis lines to translation gizmo (19ce11d)
- add bloom component action for viewing scene bloom (a4bc32e)
- add context menu option to jump to target property for property references in actor editor (7ffe128)
- add context-menu option for going to driver properties of constraints in actor editor (75a804a)
- add feedback messages when shader kernels are being built (5f9db8f)
- add film clip context menu options for editing film clip properties (d2162bb)
- add gui.PFMViewport:ScaleSelectedActors (d84df2a)
- add hover text for selected bone in viewport (9674035)
- add interactive skeleton visualization (235e50c)
- add menu option to convert scenebuild of current project to map (de73ba9)
- add menu option to start lua debugger server (bd6e21e)
- add meta type information for transform properties of pfm_actor component (23bd079)

- add model explorer option to convert model from/to ascii/binary format (162f983)
- add mouse hover selection outline effect (8a5754b)
- add option to create "child_of" constraint through actor editor (10d4a8d)
- add option to remove drivers and constraints via icon in actor editor (4bebe88)
- add outline effect component (3ea6fe5)
- add PFMTreeViewElement:AddUniqueIcon and :ClearIcons (70bcef9)
- add saving / loading to ik rig editor (7348c27)
- add support for property reference component property types (49805d2)
- add support for read-only component properties in actor editor (1456594)
- add support for very large render resolutions (0ea820c)
- add tiled rendering for pragma renderer (0f19281)
- add tutorial system (a35ca15)
- add update-checker (8d965e4)
- add utility functions and identifiers for tutorial system (6030ef5)
- add vr_manager component (wip) (b411791)
- add world axis gizmo (52ba356)
- adding a constraint or math expression now automatically adds a single-value animation channel if one doesn't exist (c44eadc)
- all logging categories are now enabled by default if log parameter is not specified (ba01a4d)
- animation drivers can now be created through actor editor (eb05de4)
- **anim:** bones can now be selected by clicking a triangle weighted to that bone (1416fbb)
- **anim:** non-selected IK bones can now be directly clicked and moved in viewport (ea6f8d5)
- **anim:** selected bones are now highlighted (b465641)
- apply changes for Silverlan/pragma@e6e7bb59 (ab4cfcd)
- **asset,gui:** add missing gui material files for detaching windows (a353b92)
- bones in actor editor are now displayed in hierarchical order (ca59b1c)
- camera settings are now saved and restored to/from projects (132c05d)
- change implementation of util_transform component to allow usage in secondary scenes/viewports (df35060)
- component lists in actor editor now use localized component names (569e80c)
- component properties with a single animation value can now be edited directly outside of graph editor (caca401)
- components in actor editor are now listed directly below actors instead of "components" sub-item (703fa63)

- converting map to actors now takes scale into account (e0e5d9a)
- copying the property path of a property now includes the full path including actor id (d41ff17)
- current project file name is now displayed in window title bar (363e53d)
- **dev:** add option to assign actor component to x / y variable via right-click context menu in actor editor if developer mode is enabled (3a22890)
- disable bloom for transform gizmos (58e4438)
- disable motion editor button (3e8f00f)
- **ecs:** add option to specify ik rig from file for ik_solver component (ad57536)
- **ecs:** add pfm_project_manager component for creating PFM scenes outside of PFM (f43f679)
- **ecs:** remove ik_solver component (5f7fc65)
- emission_factor for cycles pbr shader can now be used without emission texture (3e156cd)
- enum-based component properties no longer list "Count" as an option in actor editor (6f9f12f)
- **gui:** add ik rig editor (8d5dffc)
- **gui:** add option to add simple non-editable text properties to control menues (6c1c3e5)
- hide render options not supported by selected renderer (2edcc14)
- ik rig editor updates and improvements (0b92277)
- **ik:** add ik components (wip) (a63e07f)
- **ik:** increase precision of ik solver (90d4081)
- implement changes for Silverlan/pragma@0c91c31c (4668b27)
- importing asset into model explorer now gives an option to import as collection or as single model (55b0565)
- importing assets via model explorer now imports into sub-directory (e32e3ee)
- improve skeletal bone visualization in viewport (0315c90)
- **locale:** add ik rig editor localizations (758c199)
- **locale:** add Japanese translation (2c9fa7b)
- **locale:** add localization for components and component properties in actor editor (cfc0579)
- **locale:** add missing localizations (9115eec)
- **locale:** add missing localizations (a464a92)
- **locale:** update localizations (dfde1a8)
- math expression window now gives immediate feedback about validity of expression (88cfd5d)

- model browser now automatically refreshes after importing assets (9de100f)
- moving actor between collections is now instantaneous (50c2ff8)
- PFM projects can now be marked as read-only (0e423a5)
- pfm projects can now be saved/loaded in ascii format (4647fc2)
- pfm_actor component now also adds origin component (a6793d8)
- placing an actor through the model catalog now assigns the model name to the actor's name (f71b3f3)
- properties in actor editor now only display the base name of the property instead of the full name (d31eb26)
- remove origin component from pfm_actor component (e8f2679)
- rename "track_to" constraint to "look_at" (see Silverlan/pragma@da75b3f6) (aab83b5)
- replace selection wireframe with outline effect (d0fb7f5)
- selecting an actor in the viewport now automatically scrolls to it in the actor editor (c5e5363)
- sfm projects can now be imported from directories other than "elements/sessions/" (f41b93c)
- **shaders:** add missing particle shader files (d786f7e)
- tutorial layout changes (7c4e058)
- UDM editor now has options for creating new UDM files, importing UDM data or saving as new file (1789cce)
- various actor editor additions (74ac3b7)
- various ik rig editor additions and improvements (3019232)
- various tutorial system improvements (c0c97e9)
- version info can now be clicked to copy it to clipboard (1ad002f)
- **vr:** vr manager now automatically creates actors for all recognized tracked devices (1add829)

Bug Fixes

- add missing include (2d00d50)
- constraint menu no longer shows up for string-based properties (73fcbc5)
- debug objects are now longer included in renders when using Pragma renderer (edbdf52)
- fix "Save As" not updating internal file name to new file name (e529ab0)
- fix ascii-based models not appearing in model explorer (fda5dcd)
- fix clicking mesh not detecting correct weighted bone properly (864004e)

- fix component property values of type 'entity' not getting saved/loaded properly to/from a PFM project (68ca36c)
- fix device type settings in render options being inverted (fc4f53d)
- fix gui.PFMTreeViewElement:SetIdentifier not registering identifier with parent (3f18518)
- fix importing map not importing prop_dynamic entities (a446d9c)
- fix incorrect class name for pfm_project_manager component (1dfd98c)
- fix light sources not showing up in secondary viewports (8118eb8)
- fix low-resolution preview render image (c0f888d)
- fix lua error when auto-saving (0e818e4)
- fix lua error when baking lightmaps (84a714c)
- fix lua error when changing component property values (1b55a9f)
- fix lua error when changing property values (7d7492a)
- fix lua error when clicking ik handle in viewport (25c38e2)
- fix Lua error when launching PFM (5baf87d)
- fix Lua error when moving actor to empty collection (b71a946)
- fix lua error when rendering if opencv module is not installed (ab71e1e)
- fix lua errors (8bec2e8)
- fix lua errors if actor is spawned with unknown component types (8bf3ff3)
- fix Lua errors in ik rig editor (deb7eb2)
- fix lua errors when adding or removing constraints (70dddfe)
- fix potential crash when selecting or de-selecting spot-light actor (a3bc308)
- fix quick-ik in actor editor not working properly (68a190b)
- fix removing pfm_skeleton component not removing visual bones (3ceb0c9)
- fix some items in actor editor tree having incorrect identifier assigned to them (1288805)
- **ik:** fix broken ik behavior for actors without the pfm_model component (8ed45f6)
- **locale:** fix changing language not being saved (e997b2d)
- resolve issue where actor editor component icons would disappear in some cases (f2bd880)
- resolve issue where actor editor component property list would not get reloaded when component properties change (a81095c)
- resolve issue where camera projection matrix would not be reset properly after tiled rendering (aebe5c8)
- resolve issue where clicking constraint icon in actor editor would not select associated property properly (1226112)

- resolve issue where incorrect slider step size would be used for slider UI elements (988bea9)
- resolve issue where removing an actor could result in corrupted actor references (6f9ad31)
- resolve issue where sky actor would appear with error texture (5ba11fc)
- resolve issue where transform gizmo arrows would not transform in correct space (41b7ee3)
- resolve issue with actor model being initialized too late, causing dependent component property values to not be initialized properly (fa03294)
- resolve various issues caused by filmmaker being initialized before game is ready (d690b82)
- resolve various issues related to transform gizmo (1667afe)

Pragma (23-03-31)

Features

- "scale" keyvalue for prop and transform components now accepts vector values (755e0cd)
- add additional error reporting if model loading or importing fails (e0853ac)
- add base editor UI element identifiers (2fc6c4d)
- add bloom component properties (56042d1)
- add console commands log_level_console and log_level_file for changing log levels on the fly (8a6becd)
- add error handling when saving a model (cd7524d)
- add functions for getting parent pose of transform-based component properties (e4c108b)
- add git sha to version information (7b34f3c)
- add global string table (6f67c6a)
- add gltf import support for multiple mesh instantiations (b70ee2a)
- add gui.WIBaseEditor:GetContentsElement (31c18be)
- add logging messages for lightmap system (6d7813e)

- add missing log_file launch parameter (15c0e4c)
- add model flag for indicating if model is world geometry (75930c4)
- add Model::RemoveUnusedMaterialReferences (91bfd95)
- add new logging system (7d8c965)
- add on-complete callback to util.import_assets (1ab7ffd)
- add option to calculate projection matrix for tiled rendering (d862e63)
- add ostream overloads for entity reference types (10554e5)
- add support for importing gltf scene as map with multiple models (802061e)
- add support for Lua development and debugging with Visual Studio Code (405dabf)
- add udm lua bindings (b66c082)
- add udm.BaseSchemaType:ReloadUdmData (43357c3)
- add util::generate_two_pass_gaussian_blur_coefficients (f1c9c92)
- add util::to_image_buffer overload for providing a pre-existing staging image (00397a6)
- add WorldData Lua bindings (68eaa49)
- **anim:** add multi-threading for animations (36029e2)
- **anim:** implement constraint influences (ec65ae4)
- **asset:** add skybox materials (11fa969)
- **asset:** replace toolsnodraw texture (3b18acc)
- code cleanup (e4ad13f)
- debug_aim_info command now attempts bhv raycast before physics raycast (6c27acb)
- **debug:** add info.txt to crashdump (955642d)
- **debug:** add logging information (74977af)
- **debug:** crash dump now includes vendor information about all available GPUs on system (772db85)
- **debug:** crashdump now includes device id for available GPU devices (44fde2e)
- **debug:** re-enable crashdump exception information (b49bdaf)
- **ecs,animation:** add animation constraint system (e628639)
- **ecs,lua:** add option to specify type meta data as well as custom setters and getters for Lua-based component properties (8222143)
- **ecs,lua:** lua-based component properties using the PoseTypeMetaData type meta data are now automatically associated with respective pos/rot/scale properties (0c91c31)
- **ecs:** add "pose" property to ik solver component (07b37ee)
- **ecs:** add BaseBVHComponent::GetVertex (a62a322)
- **ecs:** add child_of constraint component (679f76e)
- **ecs:** add component property meta type data (fed2d66)

- **ecs:** add component property parent meta type (2d8f34f)
- **ecs:** add constraint component events when driver or driven object has changed (88301a7)
- **ecs:** add constraint entity component types (1410021)
- **ecs:** add event callback when clearing material overrides for model component (904fa93)
- **ecs:** add ik_solver component (a69e632)
- **ecs:** add methods for retrieving coordinate-based component properties in specific coordinate space (b26c539)
- **ecs:** add origin entity component (e9b5029)
- **ecs:** add pose type meta data for associated pose properties with pos/rot/scale properties (8b24818)
- **ecs:** add property component member type (5bde6fc)
- **ecs:** add read-only member property flag (a08a204)
- **ecs:** add track_to constraint component (06001d5)
- **ecs:** BaseGenericComponent::EVENT_ON_MEMBERS_CHANGED now includes event data struct with component information (23f75d9)
- **ecs:** implement constraints: limit_location, limit_rotation, limit_scale (34a3b26)
- **ecs:** improve performance of composite component (08aaacb)
- **ecs:** rework and rename track_to constraint to look_at constraint (da75b3f)
- emission_factor material property can now be used without emission map (39fa66c)
- Engine::AddTickEvent is now thread-safe (c00bd5d)
- entity model bodygroups are no longer reset on spawn (16c1177)
- **ExprTk:** update to version 0.0.2 (e354bb2)
- force panima animations to always update, even if timestamp hasn't changed (6f23fb2)
- **gui,lua:** add gui.ScrollContainer:GetWrapperElement, :ScrollToElement* functions (4156f14)
- **gui:** add option to disable automatic rendering when setting scene for WIViewport GUI element (21c9131)
- **gui:** add option to disable menu item (489659b)
- **ik:** add full-body ik system (8455946)
- improve bloom effect (17c4e54)
- improve lightmap quality (1cfd4d2)
- **lad:** add math alias definitions to meta lua documentation (810844b)
- **lad:** add math alias definitions to meta lua documentation (fd4b7b5)
- **lad:** update lua meta documentation files (d1326d9)

- lightmap data cache component now initializes data cache automatically (6bdbe84)
- **locale:** add debug logging messages (bd52a4a)
- **locale:** add Japanese translation (ee424f3)
- **locale:** add localization for components and component properties (b6b1d40)
- **locale:** update localizations (31767f6)
- log system improvements (79a12ff)
- loggers are now flushed whenever error or critical messages are printed (6cfcb81)
- **log:** info messages now have the "[info]" prefix if printed using the logging system (9113847)
- **log:** warning, error and critical messages are now completely colored in (baede6a)
- lua-based shaders no longer re-initialize material descriptor set when it is not necessary to do so (c199a10)
- **lua,ecs:** add component property reference bindings (228dab7)
- **lua:** add console.invoke_change_callbacks binding (c568ec2)
- **lua:** add constraint function bindings (727389f)
- **lua:** add ents.AnimatedBvhComponent:RebuildAnimatedBvh binding (dc377f4)
- **lua:** add ents.AnimatedComponent:GetEffectiveBoneTransforms binding (d53c570)
- **lua:** add ents.EntityComponent:InvokeElementMemberChangeCallback binding (6af95ce)
- **lua:** add ents.is_member_type_animatable binding (51a2271)
- **lua:** add ents.parse_uri and ents.create_uri bindings (d69df6f)
- **lua:** add equality operator for prosper.Window objects (3365a0a)
- **lua:** add game.is_game_initialized, game.is_game_ready, game.is_map_initialized (f3744d7)
- **lua:** add gui.get_element_under_cursor overload with base element parameter (2589c61)
- **lua:** add ik rig lua bindings (5b7735d)
- **lua:** add Lua bindings for log library (c536f87)
- **lua:** add math.Transform constructor with ScaledTransform argument (c3ef7ed)
- **lua:** add missing entity component type meta data bindings (e655387)
- **lua:** add model, physics and misc bindings (807dc8c)
- **lua:** add options for custom push constants and material data initialization for Lua-based shaders (03c9f4c)
- **lua:** add panima function bindings for merging channels and animations (521991c)
- **lua:** add panima.Channel:TestValueExpression (8705de2)
- **lua:** add parameter to asset.import_model and asset.import_gltf for importing model as collection or single model (2038c26)

- **lua:** add prosper.Window:GetWindowTitle (9c91196)
- **lua:** add string.snake_case_to_camel_case and string.camel_case_to_snake_case bindings (273feb0)
- **lua:** add support for displaying object types when debugging with lua-debug (e5a474b), closes /github.com/actboy168/lua-debug/issues/237#issuecomment-1477469263
- **lua:** add support for displaying object values when debugging with lua-debug (2451782)
- **lua:** add udm.is_convertible binding (9535e18)
- **lua:** add util.ImageBuffer:Insert bindings (7cc2ff8)
- **lua:** ents.Entity:FindMemberInfo now also returns component reference (8018ad9)
- **lua:** rename ik.Solver:RemoveDragControl to ik.Solver:RemoveControl (9f498d7)
- map format can now contain entity component data (de913ce)
- performance improvements (71fb75c)
- reduce weight of directional light sources when calculating dominant light direction (b935e2b)
- remove ambient color from shader render settings and add camera fov (e6e7bb5)
- update intel vtune path references to version 2023.0.0 (812e0f7)
- update panima submodule (04c72ec)
- update sharedutils submodule (7d0e338)
- update sharedutils submodule (048f8a4)
- update sharedutils submodule (8476465)
- update util_image submodule (8b48504)
- various animation driver updates and improvements (80953a8)
- **windows:** change console font to "Ubuntu Mono" (f290627)

Bug Fixes

- add missing include (c16c6d1)
- add missing origin component source code files (c1ad5eb)
- changing collisions of a physics constraint having no effect (e76ac98)
- clang compile error (b147489)
- crash when doing mouse or keyboard input before main window has been created (4855962)
- **ecs:** add missing component property registration for bone pose property of animated component (edb3798)
- **ecs:** fix enabling or disabling axis for child-of constraint not having any effect (362dbed)

- **ecs:** fix Lua errors in lua_script component (343ec34)
- **ecs:** fix lua errors when loading lua_script entity component (a32a0d4)
- **ecs:** fix space constraint component axes being inverted by default (83fae4a)
- **ecs:** resolve various ik_solver component issues (9ab5716)
- **ecs:** resolve various ik_solver component issues (5931047)
- fix "color" keyvalue for color component being interpreted incorrectly (4a3faa4)
- fix "debug_hdr_bloom" console command not working properly (a5ceec4)
- fix accessing invalid iterator when removing entity component from within OnRemove (6c60c49)
- fix BaseEntityComponent::SetTransformMember* functions applying value incorrectly if value is in world space and property expects value to be in local space (79a0a2c)
- fix build error (9c5ff63)
- fix build error (313436a)
- fix build error on latest Visual Studio version (20b1ad8)
- fix clang compile error (3b493af)
- fix ClickComponent.raycast not applying filter correctly to static entities (cfbfa5b)
- fix component properties not being saved when saving map data (2a91492)
- fix component property references not working if property path has multiple path components (41bb680)
- fix crash when removing entity with panima component (30e029e)
- fix importing gltf as model not applying mesh scaling correctly (87cf3cc)
- fix importing gltf as multiple models resulting in invalid bodygroup references (3993611)
- fix invalid memory read access (8e2c045)
- fix limit location/rotation/scale components not working if no driver was specified (17362ef)
- fix limit_distance component not working properly (fcc60fb)
- fix memory corruption issue (9dd303f)
- fix ModelSubMesh::SetVertexWeight using wrong vertex weight set (3a30e1b)
- fix obsolete model path (cb063d2)
- fix potential crash when loading a map (19fa263)
- fix potential exception in BaseEntity::CreateMemberReference (1ce36ac)
- fix potential null pointer access (6ddb2c6)
- fix pragma::get_component_member_name_hash and pragma::get_normalized_component_member_name returning incorrect results (d66af40)
- fix warning about missing static_bvh_cache component during map load (82479eb)

- **gui:** fix file explorer adding additional slash to path if file is in root directory (16f4def)
- **linux:** fix build errors due to missing std::format library (5a25f4d)
- **linux:** resolve compile errors (608da84)
- **log:** resolve various logging inconsistencies (0682925)
- **lua,gui:** resolve issue where "OnTextChanged" callback would not work for "WIText" GUI elements (ce84793)
- **lua:** add ents.create_entity_uri and ents.create_component_uri bindings (1dfac80)
- **lua:** fix duplicate bound "tostring" method for entity components (954323a)
- **lua:** fix ents.BhvComponent.HitInfo.entity not using correct lua object (1ae8906)
- **lua:** fix ents.UniversalMemberReference:GetPath not returning correct path in some cases (6bc89ff)
- **lua:** fix error when calling log.prefix (e8203c3)
- **lua:** fix incorrect constness for some lua bindings (7000a7f)
- **lua:** fix lua error when calling ents.UniversalComponentReference:GetComponentName (12ba931)
- **lua:** fix module paths not being set up properly unless remote debugging is enabled for lua state (7de40ff)
- **lua:** fix potential invalid memory access in util.generate_hair_data (7ae2ffd)
- **lua:** fix type enums for ik rig constraints being bound with wrong class (7ef9257)
- **lua:** implement missing features for transform component property type (d40c88d)
- **lua:** resolve issue where udm GetValue functions would return nil if property is of type element (5a211e8)
- **lua:** resolve issue where universal entity reference constructors would not accept uuid objects (f7b35d6)
- ModelSubMesh::GetVertexWeight - resolve issue where incorrect vertex weight set would be used (fff66d9)
- resolve compile errors (6079768)
- resolve issue where console would not get initialized fully unless created at engine startup (1165a97)
- resolve issue where crashdump would not generate properly (a0df754)
- resolve issue where debug render objects could accumulate and not be flushed when no scenes are being rendered (a9ba75e)
- resolve issue where entities in map generated from imported gltf would all have same uuid (beca3c8)
- resolve issue where entity render buffer would not be initialized properly (6b9e379)
- resolve issue where ik rigs could not be saved (8500d0a)
- resolve issue where importing a model through the model explorer would fail (e93b81c)

- resolve issue where lua source file could not be determined correctly (4ccd7ce)
- resolve issue where Lua-script for Lua-based entity would not be loaded if invoked by creation of another Lua-based entity (6feee4a)
- resolve issue where material and entity color factor would be interpreted as srgb instead of linear (6cc1778)
- resolve issue where objects from imported gltf assets would have incorrect rotation (656399a)
- resolve issue where panima components would not be updated properly if entity has no animated component (85aaa9a)
- resolve issue where saving a model would not save the reference animation correctly (be93859)
- resolve memory corruption issues (7ed7018)
- resolve potential crash when multiple scenes with debug objects are rendered at the same time (31527e2)
- resolve potential crash with animation drivers when using parameter references of properties with certain types (3f34a22)
- temporarily disable multi-threading for animations (524ad34)