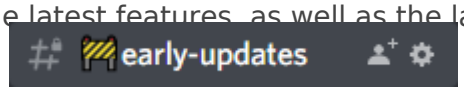
May 2022 Progress Report

GitHub Beta Releases

First things first, a few changes on how I'll be doing beta releases in the future:

I've set up automated builds via continuous integration (CI) using GitHub actions for the Pragma repository.
This means that from now on every time I commit new changes to the source code, there will be an automatically generated beta-release with those changes available on GitHub a few hours later:

https://github.com/Silverlan/pragma/releases/tag/nightly

These releases always include the latest features, as well as the latest version of the Filmmaker.
There will be a notification in the 

channel on Discord whenever a new version is available.

> Please remember that these are **beta**-versions, which means they're not necessarily stable and some things may not work properly or suddenly break in an update. If you want to be on the safe side, you can wait for the next stable release, however I don't have an eta for the next one at the moment.

If you have the Steam version with the preview beta-test DLC, you'll still get regular updates like before.

Prelewd will get a similar treatment soon, but I haven't quite finished the setup for that yet.

Linux Support

I've also been working on and off on adding Linux support to Pragma/PFM. Most of the libraries are Linux-compatible now (indicated by the 

on the GitHub repositories), but there's still a lot of work ahead for the core repository.
I'm hoping the new CI integration will make that easier for me, but it'll likely still be quite a while, since it's a low priority.

Filmmaker

Graph Editor

The graph editor is now mostly complete and functional. It supports:

- Constant, linear and bezier interpolation
- Aligned, free and vector handle types
- Same easing types and modes as Blender (Sinusoidal, Cubic, Exponential, Bounce, etc.)

Some of the graph editor buttons don't do anything yet, and there's always room for QOL changes, but the core implementation is complete. I've yet to write a wiki article on how to use it, however.

Cycles X & Improved live raytracing

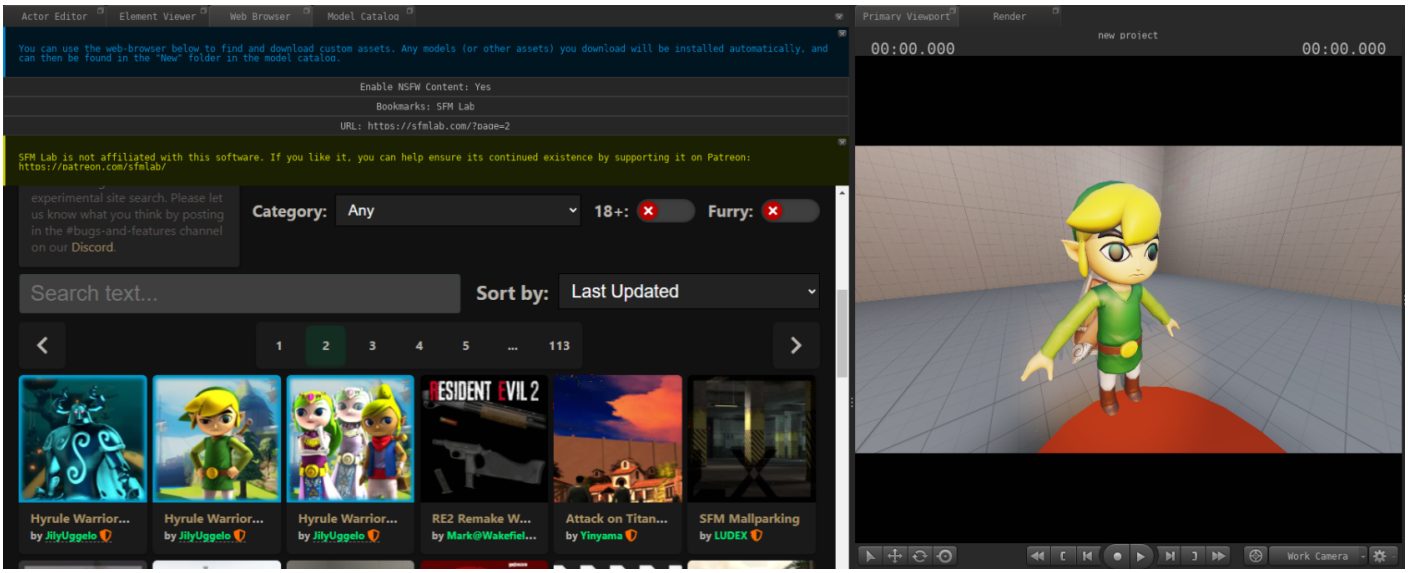
Cycles X is a new major update to the Cycles renderer that was added to Blender last year. A few months ago they also made it available as a stand-alone version, so I took some time to upgrade to it. Some of the changes include:

- There should no longer be rendering issues with certain GPU models
- Rendering should be **significantly** faster now
- Adaptive sampling is enabled by default when rendering with Cycles X in PFM, which should decrease render times even further
- The OptiX denoiser is used by default if available
- Cycles X can now be used for the live raytracing render viewport, which is almost in real-time:

You **can** edit the scene while live raytracing is enabled, but only some changes to the camera and light sources are updated automatically for the time being. Other changes don't appear in the live raytracing view until it's reloaded.

Internal Web-Browser

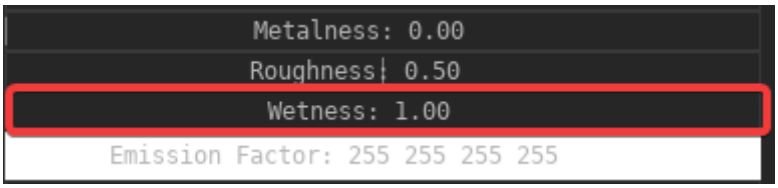
I've added an internal web-browser to the Filmmaker interface:



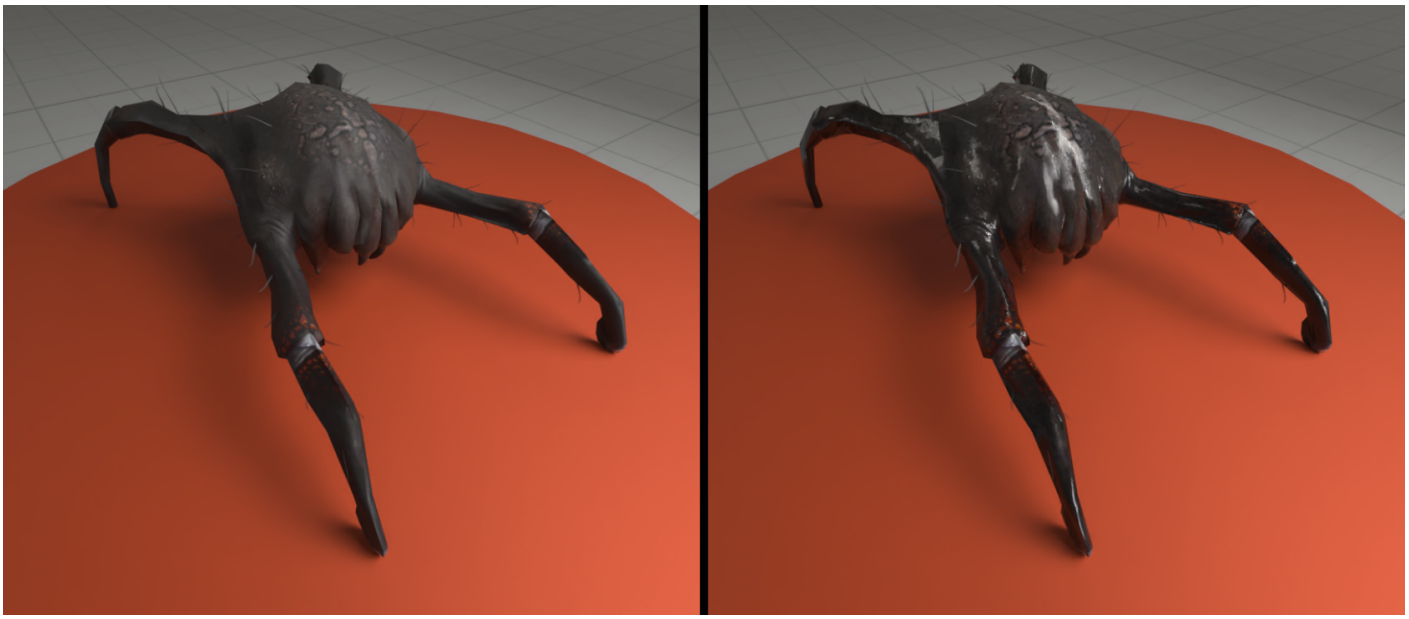
It has bookmarks to several big SFM and Blender content websites (e.g. SFMLab, SmutBase, etc.) and any files you download through it will automatically be analyzed by Pragma. If Pragma detects any asset files in the download (e.g. models, maps, etc.), they will **automatically be imported** and will be usable immediately. This works with most downloads from most websites, although some archive types (e.g. `.rar`) can cause some issues.

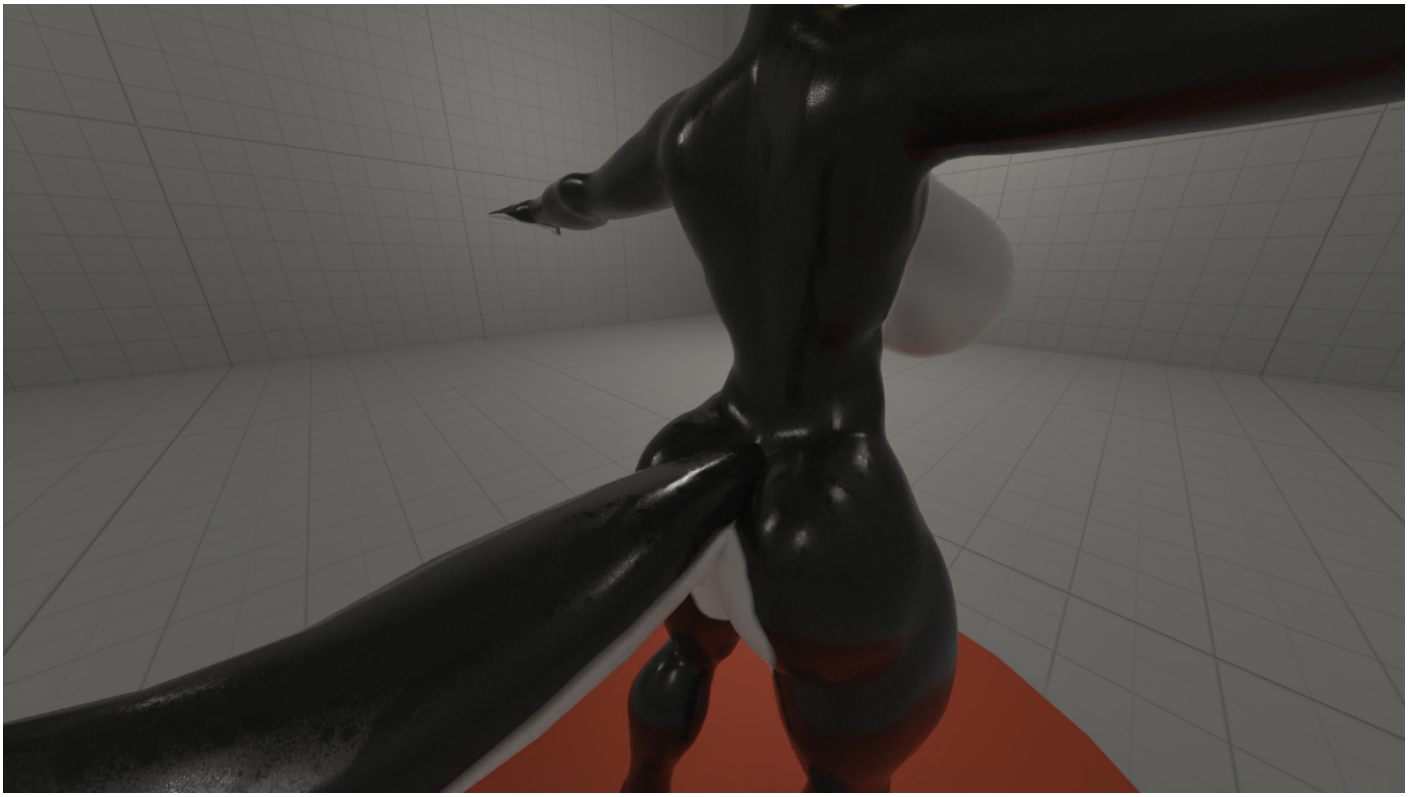
Wetness shader

I've had several requests for this, so I added a "wetness" shader for rendering with Cycles X.
To use it, all you have to do is crank up the "wetness" slider for the material you want to use it for in the material editor:



The effect only applies when rendering with Cycles X at the moment, and will create the appearance of wet skin:

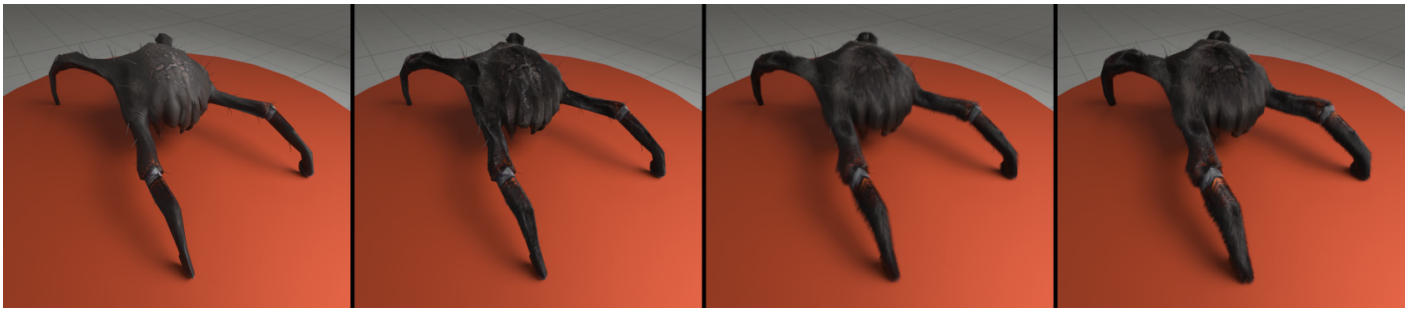




It's not perfect, but it's significantly better than to just rely on the roughness slider. If anyone has experience with Cycles and has some ideas for other neat effects, let me know, I'd be happy to implement them!

Fur / Hair rendering with Cycles X

Fur and hair rendering is now fully functional with Cycles X as well (previously only LuxCoreRender).



(From left to right: No hair → Increased levels of hair.)

This one doesn't look great with this particular model, but I don't have any other examples right now and I think you get the gist:



IK Posing

The IK posing I added a while back is now finally integrated into the filmmaker interface:

That being said, IK limits (e.g. to prevent an arm from bending more than 90 degrees) are currently not supported.

Animation with IK should also be functional, but hasn't been tested much so far.

Misc Changes

Some other changes that aren't as interesting by still worth mentioning:

Improved asset import support

Alongside the web-browser, asset import has been improved overall significantly. On top of the previously supported formats, new supported formats include: blend, fbx, dae, x3d, obj, abc, usd.

(You can find the full list here.)
Support for pmx (MMD) models has also been improved and now supports animations and morph targets.

Additionally, you can also now just drag-and-drop an entire compressed archive onto the model asset explorer and all of the assets within will be detected and imported automatically.

Detachable Windows

You can arrange your workspace however you want by detaching windows from the primary window:

This is particularly handy if you have multiple monitors available. In addition, there's also support for multiple viewports now (up to three):

Copying actors

Actors (including animation data) can now be easily copied between projects through the actor editor:

When you copy an actor, it is copied as a string to the clipboard. This string is just regular text, which means you can, for instance, send that string to someone else (via chat), and they can then just use it to insert the actor into a project on their side. This makes sharing actors/animations with other people significantly easier.

Controlling light sources using work camera

Spot light-sources can be controlled using the work camera, just like in SFM:

New PFM project format

PFM has a new project file format, the old format is no longer supported.
Without going into too much detail, the new format is easier to work with and significantly faster for loading and saving. There is also an auto-save system now that automatically saves the project every 10 minutes, so you don't lose too much progress in case of a crash.

Rebindable keybinds

PFM keybinds can now be remapped via the "pfm_bind" and "pfm_unbind" console commands. There will be an interface for that (like the one in SFM) in the future as well, but for now it's console only.

Engine Updates

Support for high-poly Meshes

The vertex count of meshes was previously limited to **65535** vertices per mesh. The limit has been increased to **1872457** vertices, which should cover the vast majority of models.

This means that models with a high mesh complexity that weren't working before, should now be fully supported. This mostly relates to some Source 2 and Blender models, since Source 1 models have significantly fewer vertices.

Support for the Source Engine "Eyes" shader

Not a major update, but I've finally added support for the old "eyes" shader from the Source Engine.

Basically, the Source Engine has two different ways of rendering eyes, an older version ("Eyes", used by older Source games, like Half-Life 2, as well as a lot of custom models), and a newer version ("EyeRefract", used by TF2 and newer Source games). Previously there was only support for the new shader and not the old one, which meant that the eyes of characters using the old

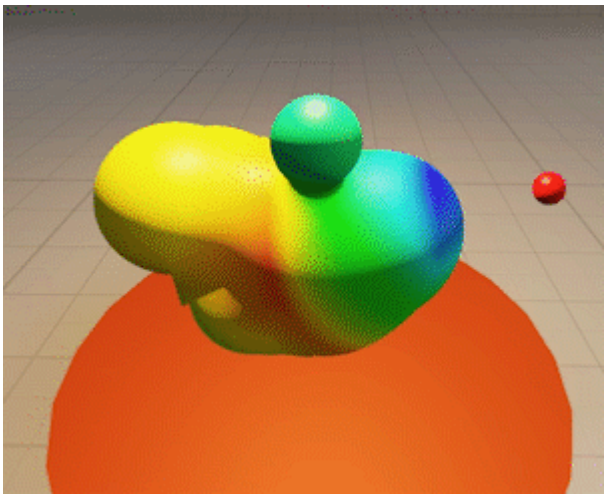
er is now fully supported as well:

Fixes

I've fixed and upgraded up a bunch of old Engine systems that have broken over time due to one reason or another. Some of these include:

Particles

Particle rendering is (mostly) functional again, and blob particles have received some updates (like support for PBR properties (roughness, metalness, ...)):



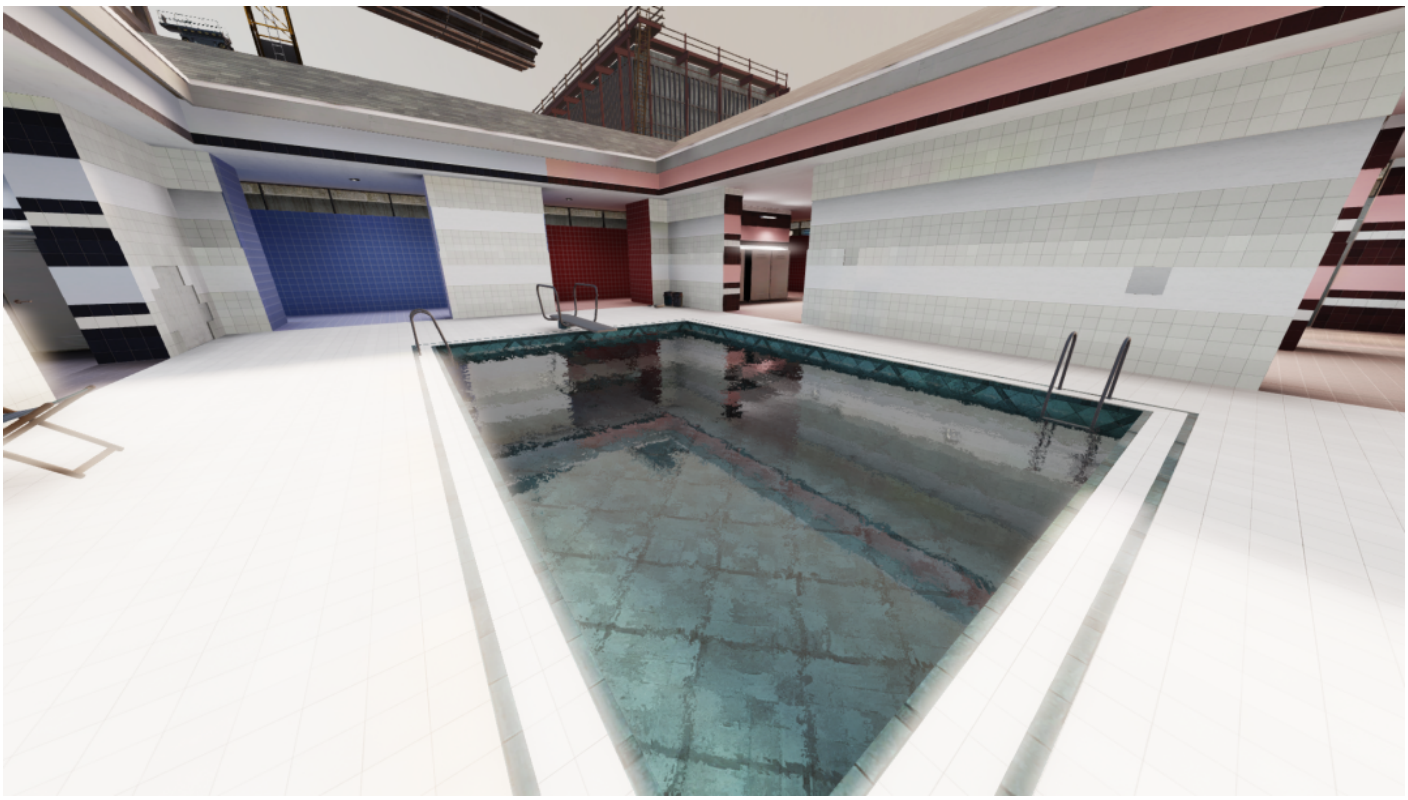
Skies

Skies now support equirectangular HDRI textures (previously only cubemaps) and can be rotated and animated:

Water

Water rendering, buoyancy physics, fog, 3d skies and portals are fully functional again:





New asset manager system

I've completely overhauled the engine's asset management system(s). I don't want to bore anyone with the details, but the most important change is that it's now fully multi-threaded, which means that asset loading (in particular models and textures) should be significantly faster now, especially for projects or maps with a lot of models.

Prelewd

Unfortunately there's still no new version of Prelewd yet, but I do have some exciting news for the near future!
I've recently acquired the usage rights to a large amount of high quality VR SFM and Blender animations by **MetsSFM** (a retired SFM animator). I haven't had the time to integrate them into Prelewd yet, but I will get started on that later next month.

I have also set up an automated build and release system for Prelewd as well, but that isn't active yet. Once it's active there will be regular notifications on the Prelewd Discord server about new pre-releases (hopefully starting later next month as well).

Other than that, there have been some UI overhauls and performance improvements, but I can't show those off quite yet. Some of the new PFM changes (like the asset web browser) will also be part of the next Prelewd update!

Full Changelog

(For previous versions, see https://wiki.pragma-engine.com/books/pragma-engine/page/changelog)

Filmmaker v0.4.6

- 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

---

Revision #67
Created 2022-03-31 14:08:51 UTC by Silverlan
Updated 2025-01-19 14:13:14 UTC by Silverlan