August 2022 Progress Report

Pragma Renderer

Next to Cycles X and LuxCoreRender, I've now added Pragma as a renderer option. Unsurprisingly, this renderer uses Pragma's internal rendering system. The main purpose for the Pragma renderer is for **animations**, since render times with Cycles X and LuxCoreRender are excessively high for large image sequences, even on modern hardware (days, weeks or even months depending on hardware, scene complexity and framerate).

To make Pragma actually viable as a renderer, I've added some new features:

Lightmapping

The lightmap system has been **significantly** improved, and now includes **directional lightmaps**. The direct and indirect components are baked with Cycles X, while the directional component is using my own baking implementation. Without going into too much detail, this (along with other improvements) allows for much prettier lightmaps than before:



For comparison, here is a pure Cycles X render of the same scene:



The pure Cycles X render still looks slightly better, but I think the baked version is a pretty damn good substitute. I am not entirely sure why the colors look more intense in the baked version, that is something I will still have to investigate.

Lightmap baking still takes some time, but the major advantage is that it only has to be done **once per scene**, regardless of how many frames you render. Obviously this only works for static lighting, but it's still a pretty major milestone. For dynamic / animated actors you would still use dynamic light sources with a baked reflection probe.

Once lightmaps are baked, the rendering with the Pragma renderer is practically real-time, so the only render time to worry about is the actual lightmap baking. Baking the lightmaps for the example scene above at a 4K resolution took about 8 hours with my GTX 1050 Ti, but a resolution of 2K would probably be sufficient for a scene like this (which would reduce the render time to ~2 hours).

Since baked lightmaps are very cheap to render during real-time playback, Prelewd will also heavily benefit from this.

Motion Blur

I've added support for camera, object and skeletal animation motion blur, however this is currently implemented for the Pragma renderer only.



(If the difference in the video is too hard to spot, try pausing it.)

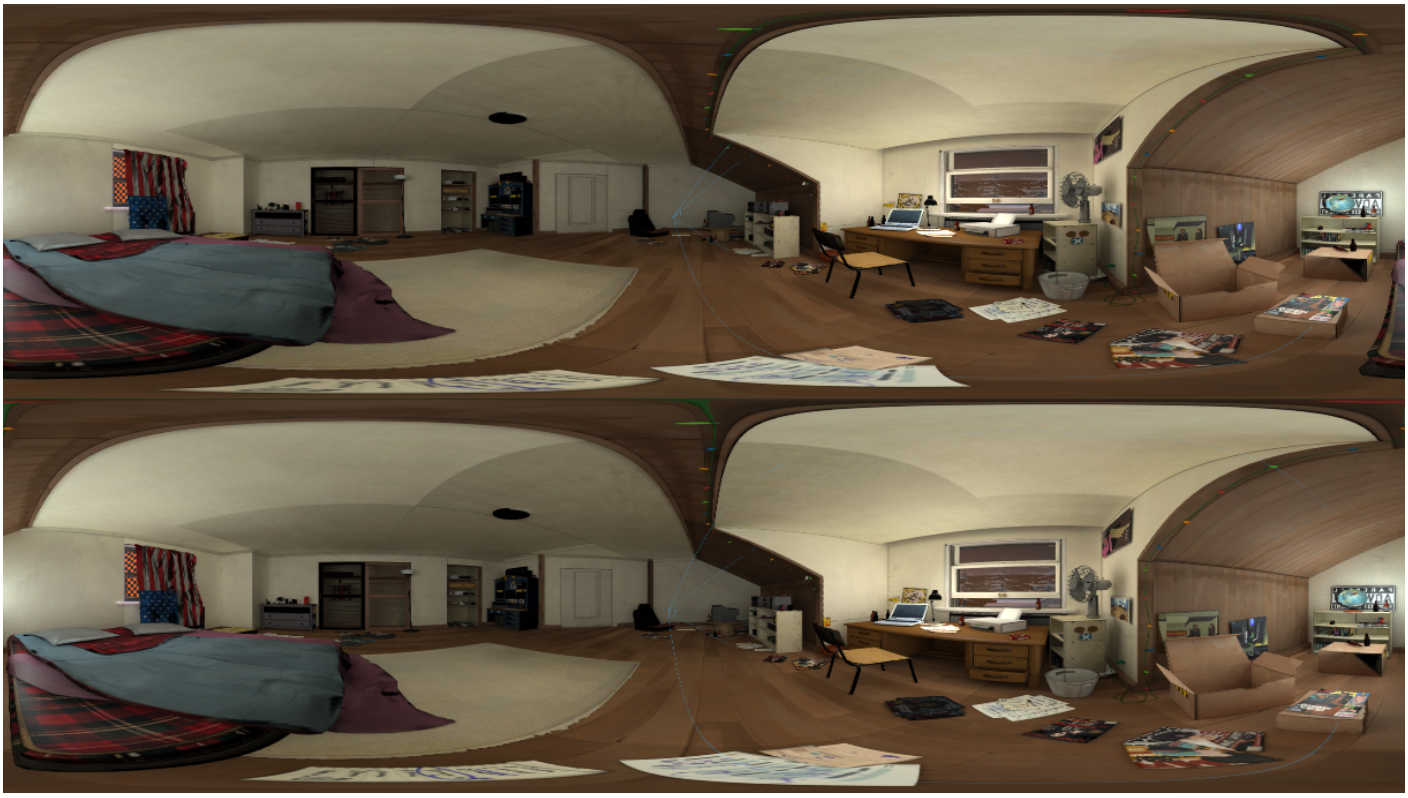
There are still some artifacts in some cases (like the Medic's finger in the right example), but it's not very noticeable in motion.

Technically the same motion blur implementation could be used for Cycles X/LuxCoreRender renders, but I don't think I'll have time to implement that any time soon, so it's restricted to the Pragma renderer for now.

Misc

Some misc features include:

- **SSAA**: The Pragma renderer uses SSAA for anti-aliasing. Looks good, but does require a fair amount of VRAM.
- **Color Correction**: You can use OpenColorIO color transforms with the Pragma renderer. Default is filmic-blender, just like for the Cycles X / LuxCoreRender renderers.
- **Virtual Reality**: Of course the Pragma renderer also has full support for rendering image sequences for VR:



New Transform Gizmo System



I've completely overhauled the transform gizmo system, since the previous one was... severely lacking in many ways. The new one is significantly more pleasant to use, and has some new features:

- Snap-to-grid (both for translations and rotations)
- Transforming multiple selected objects at once (not supported for scaling yet)
- Viewport now supports rectangle-select by clicking +dragging mouse cursor

Smart Move

There is also a new "smart" move system, which allows you to quickly move objects around while keeping their relative orientation to the environment:

Decals

Not much to say here. I already implemented decals a while ago, but I never got around to add them to PFM until now:




They can correctly wrap around most surfaces, doesn't work with animated actors at the moment though.

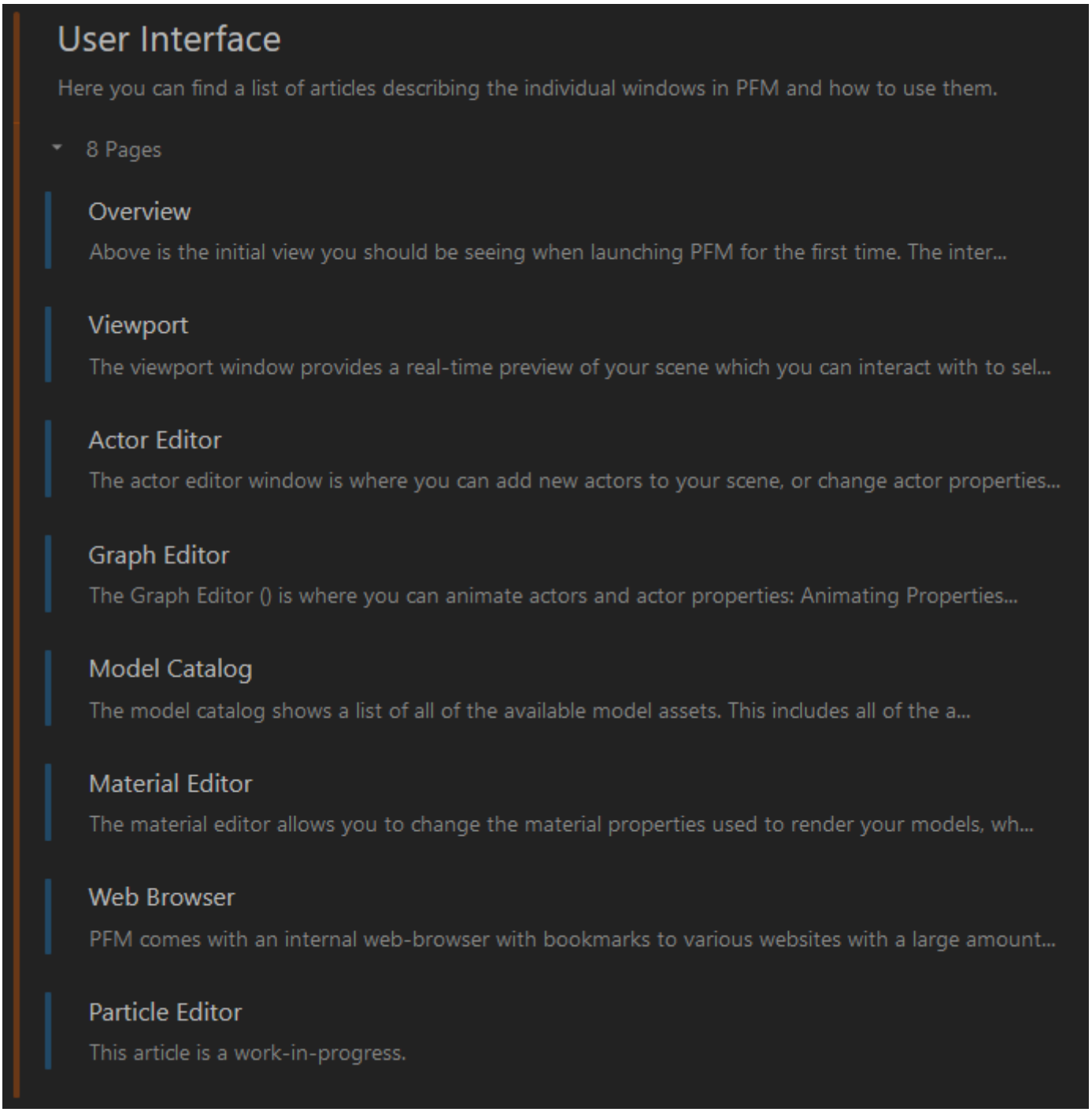
Other Changes

- **Undo-redo system:** Not much to say here other than that undoing and redoing actions is now implemented. The maximum number of undo steps can be controlled with the "pfm_max_undo_steps" console command. Some actions (like adding new actors or components) don't have undo support yet.
- **Map Import:** You can "import" a map, which adds all of the map's props (and the world), light sources, etc. as actors to the current project. You can use this if you want to use parts of different maps in the same project, for example.
- **Improved viewport interaction:** I've added an object BVH system for fast raycasts. Effectively this means that selecting actors in the viewport is significantly faster and more accurate now.
- Various UI improvements and fixes
- Lots of general bug fixes and minor improvements

Tutorials

I've added a bunch of new tutorials on the wiki:

User Interface


User Interface
Here you can find a list of articles describing the individual windows in PFM and how to use them.
8 Pages
Overview
Above is the initial view you should be seeing when launching PFM for the first time. The inter...
Viewport
The viewport window provides a real-time preview of your scene which you can interact with to sel...
Actor Editor
The actor editor window is where you can add new actors to your scene, or change actor properties...
Graph Editor
The Graph Editor () is where you can animate actors and actor properties: Animating Properties...
Model Catalog
The model catalog shows a list of all of the available model assets. This includes all of the a...
Material Editor
The material editor allows you to change the material properties used to render your models, wh...
Web Browser
PFM comes with an internal web-browser with bookmarks to various websites with a large amount...
Particle Editor
This article is a work-in-progress.

Rendering

Rendering

How to render your project and how to use specific rendering effects (motion blur, fog, etc.).

13 Pages

Rendering

To render your scene, switch to the "Render" tab above the viewport and you should see a window s...

Depth of Field

Depth of Field is currently only available for the Pragma renderer. To add a depth of field ef...

Motion Blur

Motion blur is currently only available for the Pragma renderer. PFM supports both per-object ...

Fog

Fog is currently only available for the Pragma renderer. To get a similar effect with Cycles/LuxC...

Volumetric Lighting

Volumetric lighting can be used to create fog, light beams, dust in the air and similar effects...

Virtual Reality

PFM allows you to easily render image sequences for VR videos: (Click and drag your left mouse...

HDRI Skies

If you want to use a sky in your scene, you can do so by using a HDRI sky texture. PFM already sh...

Pragma Renderer

Please see the Rendering Animations workflow for information on how to use the Pragma renderer.

Live Preview

If you're planning on using Cycles X or LuxCoreRender for rendering, you can enable the live rayt...

Animations

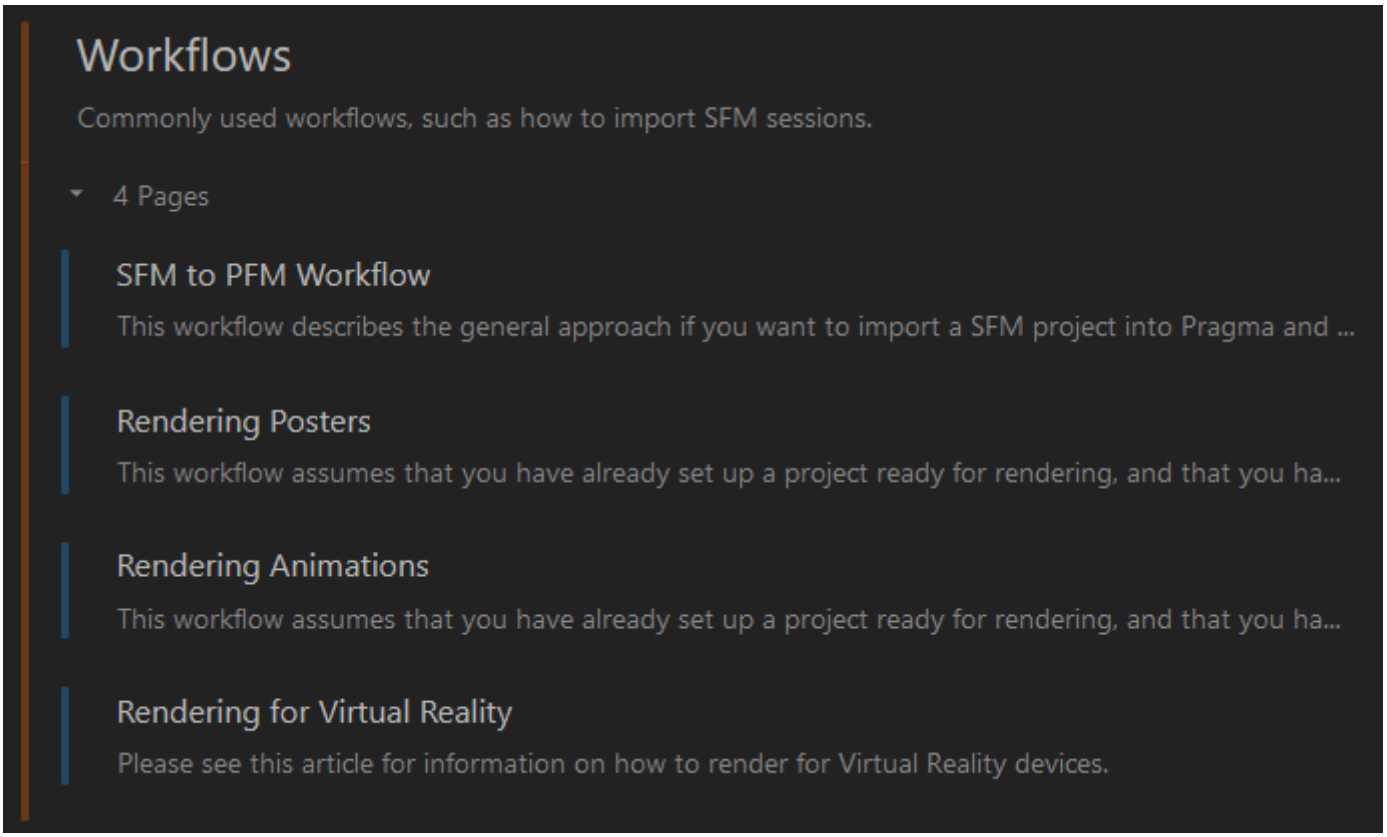
Pragma currently only supports rendering image sequences, which means you still need a video edit...

Lighting

PFM supports four light source types: Sun Lights (aka directional light or environmental light) ...

Decals

Workflows



Some are more complete than others, but Getting Started is a good starting point. The Rendering Animations workflow describes how to effectively use the new lightmap system.

Going Forward

With that I unfortunately also have some bad news. In part thanks due to the current economic crisis, I can't actually afford to continue working on this project for much longer, unless I can somehow get significantly more supporters on board □□. For that end I will have to make some changes going forward..

Primarily this means that, starting next month, **Prelewd** will become my main priority. I know that's hardly ideal considering progress on the Filmmaker/Engine has already been pretty slow, but sadly I just don't have any other options at the moment. If I can use Prelewd to reach some of my milestones, I may be able to put more time into the project as a whole again, but for now that's somewhat up in the air. With prices absolutely skyrocketing over here, and things likely becoming worse come winter, unfortunately this is a compromise I have to make at this point □□. I will continue to squeeze in as much time as I can into the Filmmaker and the Engine, but there will likely not be any *major* new features for either for some time, at least none that don't also benefit Prelewd in some way. I'm sorry, but I hope you understand.

Full Changelog

Latest release is available on GitHub: https://github.com/Silverlan/pragma/releases/tag/nightly

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

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

---

Revision #36
Created 2022-08-28 08:27:31 UTC by Silverlan
Updated 2025-01-19 14:13:14 UTC by Silverlan