

江文韬

电话: 18074927532 | 邮箱: jiangwt2001@qq.com



教育经历

桂林电子科技大学

计算机科学与技术 本科 计算机与信息安全学院

2020年09月 - 2024年06月

桂林

工作经历

厦门勇仕网络技术股份有限公司

技术美术(程序向)

工作职责:

- 渲染开发**: 参与项目中角色shader, 场景渲染, 特效等内容开发
- 预研探索**: 针对项目需求, 尝试探索落地相关前沿渲染图形技术提高渲染效果。
- 美术工具改进**: 参与项目美术工具链内容开发, 提高美术工作效率。
- 分析竞品游戏**: 针对项目目前存在的痛点, 分析有关竞品游戏效果, 考察复现相关渲染技术方案。

2024年07月 - 至今

厦门

北京君合创想科技发展有限公司

引擎开发(实习生)

工作职责:

- 引擎资产模块**: 参与公司自研渲染引擎场景资产扩展功能开发。
- 引擎渲染开发**: 修改引擎的渲染层增加特性支持。

2023年06月 - 2023年09月

北京

专业技能

- 具有多种编程语言的使用经验, 能够快速上手一门新语言并在项目中使用。了解C++核心特性, 了解Rust语言与特有机制, 了解C#在Unity中的开发应用。
- 了解计算机图形学基本概念与相关实时渲染技术如风格化渲染, PBR workflow, BRDF模型与后处理技术。
- 了解URP渲染管线工作流程, 具有渲染管线开发经验, 具有实际项目中渲染管线的迭代与优化经历。
- 了解现代图形API如Vulkan, 具有Vulkan开发经验。定位解决过移动端OpenGL ES 3.0上特定实现导致的渲染错误和GLSL编译器优化导致的报错的经验。
- 能够编写Shader实现图形渲染效果, 对移动端渲染机制和设备有所了解, 能够在结合相关硬件平台Profiler结果下把控相关效果与性能的平衡。
- 了解图形程序的调试和故障定位, 能够利用Nsight, Intel GPA, RenderDoc等分析工具对图形程序进行调试, 具有对竞品游戏渲染效果逆向经验。
- 具有良好的英文阅读能力, 能够阅读相关领域文档与文献, 持续阅读了解图形渲染领域相关分享如GDC, NVIDIA Developer。

项目经历

厦门勇仕-深空之眼(上线项目)

技术美术(程序向)

项目背景:

上线项目: 基于Unity开发的动作向二游, 渲染风格为风格化PBR流程。主要为大型箱庭场景和实时动作战斗, 运行平台覆盖移动端Android, IOS, PC端。

UI相关特效功能:

- 白噪声功能**: 基于柏林噪声和UI遮罩绘制UI底部的白噪声效果。
- 3D UI模型**: 通过脚本修改UI canvasRenderer Mesh为UI接入了3D模型。

角色相关特效:

- 角色材质接入场景级联阴影投射**: 项目原有角色不能接受场景投影, 并且项目中阴影实现方式较多, 经过梳理后, 在PC端部分室外场景下, 角色可接受场景物体投射的自投影, 并与高精度自投影混合形成较为自然的阴影效果。
- 角色遮挡描边**: 实现策划需要的在角色被场景遮挡时显示描边信息的需求, 在角色上使用基于脚本注册至单例类上, 并基于模板测试实现描边物体内部的剔除。

美术工具需求:

- 项目原有的平滑描边工具**: 基于Unity Mesh信息计算并导出具有平滑法线信息的模型, 在复杂模型上导入时间较长(3-10s左右), 改写为基于FBX SDK后, 在不牺牲美术效果的同时提高工具效率, 运行时间降低至1s内。
- AssetBundle预览工具**: 基于UGUI的Unity Assetbundle预览, 能够查看AB包内部的资产信息。

关键Bug排查

- 为了适配移动端带宽, shader计算多使用FP16半精度, 在部分机型上存在特定视角上的光照点乘结果过小, SafeNormalize出现除0导致光照后处理异常。通过截帧和反编译shader分析定位bug点修复。
- Criware视频流在升级SDK3.10版本后, 视频播放实现fallback到legacy OpenGL ES 2.0, 导致原有的通过IMaterialModifier修改UI材质输出视频做法引发视频卡顿。定位后与前端修改视频UI配置方法与参数设置脚本修改解决问题。

P10在研项目

程序向技术美术

2024年07月 - 至今

项目背景:

在研项目，主要风格为卡通箱庭大世界，风格类似Relink。

主要工作：

渲染效果相关：

1. 角色自投影：参考Unreal引擎的PerObjectShadow与Unity的Decal贴花，在不破坏管线工作流的前提下，通过对角色单独投射高精度阴影贴图，在屏幕空间采样提高角色自投影的精度。
2. GT Tonemapping实现：移植了GT Tonemapping，并根据特效需求进行相关修改调优。
3. 自定义Bloom效果实现：通过修改Bloom的滤波核，相比URP Bloom泛光区域更大，且泛光形状为更自然的圆润形状，同时通过在上采样流程加权采样多Mipmap Level图像，提高了泛光的模糊效果与可配置性。
4. 亮度保持映射：尝试在项目落地实现了亮度保持映射(LPM)后处理效果，能够基于亮度调整画面饱和度，并适配多种显示色域。
5. 水特效实现：基于OpaqueTexture和法线贴图的UV流动动画，实现流动水的效果，并接入粒子系统通过顶点色控制水的消散。

角色需求验证相关:

1. 史莱姆怪物效果验证：基于OpaqueTexture实现了史莱姆的半透明质感和模糊效果。
2. 通用特效脚本：通过URP DrawRenderer在单独pass绘制角色特效，实现角色特效与shader实现解耦，支持rimlight，闪烁，遮挡描边，溶解等通用功能。

URP渲染Demo(个人项目)

项目简介:在URP14中实现的各种渲染效果原型Demo

项目链接:https://github.com/af8a2a/URP_Extension

1. 基于DrawRenderer与模板测试实现物体遮挡描边
2. 前向管线下实现的屏幕空间光线追踪(SSR)，支持时域累积等功能。
3. 基于后处理的体积光功能原型。
4. AMD FidelityFX-LPM亮度保持映射在URP下的实现，基于HDR下的屏幕亮度信息与映射曲线调整画面饱和度。
5. 基于柏林噪声实现的屏幕白噪声效果。
6. 半透明玻璃效果和点阵半透明。
7. Diffusion晕染后处理效果。
8. UI刮刮乐效果，在Image上记录鼠标按键移动区域绘制到单独的RT上，基于Mipmap低开销获得RT记录的刮卡进度。

北京君合创想-自研引擎开发

项目背景：

公司通过自研的渲染引擎支持数字孪生相关业务开发，支持PBR工作流渲染，底层渲染模块基于OpenGL Core，资产导入导出基于Obj文件格式，上层业务代码基于AngelScript，调用C++导出的绑定接口与底层引擎Runtime交互。

1. 原有引擎通过分析Obj文件格式对场景资产进行导入导出，每次导入时间较长影响工作效率，且不支持Unity等引擎通用的FBX格式。梳理引擎资产工作流程后，修改为基于FBX SDK，为渲染引擎的资产导入导出功能添加了FBX等格式的支持。同时针对渲染场景物体导出结果可合并的特点，使用OpenMP并行化加速渲染引擎场景导出的效率，大规模场景(100+MB)下的导出时间从5min优化至15s左右。
2. 原有引擎未适配HDR贴图格式，通过修改引擎渲染层的图形API接口，为渲染引擎增加了HDR贴图CubeMap的适配和Mipmap的支持。