

潘程

13916355957 • p1067838263@gmail.com • www.charlesrealm.com

教育经历

南加州大学, 硕士 绩点: 3.85/4.0 主修计算机科学-游戏开发	2024.09 - 2026.05
纽约大学, 本科 绩点: 3.797/4.0 主修计算机科学, 辅修游戏工程, 数学	2021.09 - 2023.12
康涅狄格大学, 本科 绩点: 3.941/4.0 主修计算机科学与工程	2019.09 - 2021.06

技能

软件: Unity3D, SVN, Sourcetree, UGit, VSCode, VS2022, Photoshop, Maya, ZBrush, Substance
编程语言: C, C#, C++, CSS, HTML, Lua, JavaScript, Python, GLSL, Shaderlab, SQL

工作经历

Unity 客户端实习生 | 上海淘米网络科技有限公司（圣然） | 《赛尔号巅峰之战》 2025.06 - 2025.09

- 使用 **UGit** 以进行项目版本管理与开发, 了解项目分支的合入流程与 CI/CD 自动化关系
- 每周交付 1-2 个活动/关卡 UI 需求, 独立完成从开发到联调的全流程, 上线前通过 QA 验证率 100%。
- 使用 **Profiler** 对图鉴点亮滚动列表功能进行性能消耗监控以进行分帧加载优化。在 **Deep Profiler** 条件下, 首开主线程尖峰由约 900 ms 降至约 700ms (↓ 200ms, 约 22%)。
- 在主程指导下走查功能上线发布流程: 从功能合并至 release 分支、CI/CD 构建产物, 到 CDN 边缘预热与分发, 再到版本号门控触发差分补丁。
- 系统性理解 **YooAsset** 与 **HybridCLR** 链路: YooAsset 的分包/版本校验/差分补丁与 CDN 分发协同。
- 了解 HybridCLR 热更原理: 通过差分更新 DLL 实现 C#代码热更, 避免重新打包; 与 YooAsset 资源热更流程协同。

Unity 客户端实习生 | 上海哟尔哈科技有限公司 | 《命令与征服: 军团》 2023.06 - 2023.09

- 使用 **SVN (Subversion)** 进行版本控制与团队协作。
- 参与项目迭代, 使用 **Lua** 开发并优化活动页面 UI, 并进行全面测试, 确保高质量输出;
- 使用 **Profiler** 对滚动列表功能进行性能消耗监控以进行分帧加载优化。在 **Deep Profiler** 条件下, 首开主线程尖峰由约 1000+ ms 降至约 700–800 ms (↓ 200–300 ms, 约 20–30%)。
- 与项目负责人、设计师和开发人员合作, 并参加每周进度会议, 以保持流畅的工作流程并及时解决新出现的问题。

Unity 实习生 | 上海完美时空软件有限公司 | 《朝与夜之国》 2022.07 - 2022.09

- 使用 **AirtestIDE + Python** 编写自动化测试脚本; 在 Unity 中以 TypeScript 进行前端功能开发
- 学习使用异步方程和第三方插件及 SDK (如 DOTween) 的封装等高级概念。
- 与项目领导和开发人员合作, 并参加每周会议, 展示进度报告, 确保与项目目标保持一致, 并及时解决问题。

项目经历

Prime Engine 功能扩展 (教授自研引擎) | C++ / Lua

2025

- 走查事件队列 (Input Event) 与广播 (General Event) 双模式分发系统，支持低延迟交互。
- 掌握 Maya 编辑关卡→Lua 配置组件数据→导出.xlevel 文件→C++解析的全流程开发。
- 引入动画权重与分层混合 (上半身射击 + 下半身跑步)，支持渐入/渐出与过渡；解决全身姿态冲突并减少动画切换伪影
- 在模型导入阶段生成并缓存 AABB 包围盒；实现视锥体剔除；用 LineRenderer 做包围盒可视化调试，降低无效绘制
- 设计 PhysicsManager / Rigidbody / Collider 组件化框架；为静态场景构建 BVH 加速结构；支持运行时拖拽更新碰撞体并触发回调更新 BVH 结构，保障移动单位与静态触发体的稳定碰撞。
- 扩展 Lua 功能：支持 Lua→C++静态方法绑定，实现脚本化行为（保留 C++主逻辑以规避解释器性能损耗）。

PBR 渲染 | C++ / OpenGL / GLSL

2024

- C++ 侧资产管线：解析 Suzanne 模型与由 Substance Painter 导出的材质贴图，创建 OpenGL 纹理并绑定采样单元；整理不同 material/roughness 的对比渲染配置
- 几何与数据上传：使用 VAO/VBO/EBO 组织顶点 (position/normal/UV)，定义属性布局；基于 GLM 计算并通过 uniform 传输 MVP 矩阵、相机位置与多光源参数。
- PBR 着色：GLSL 中实现 Cook-Torrance 框架 (GGX NDF、Smith GGX 几何项、Schlick 菲涅尔)，物理一致的能量分配。
- IBL (环境贴图照明，split-sum)：集成 irradiance map 做漫反射，prefiltered environment map + BRDF LUT 做高光反射并按 roughness 选择 MIP；与 AO、直射光结果合成最终颜色。