工程实践腾讯企业选题

邮箱：szsse@ustc.edu.cn

发送简历命名：企业名+题目编号+姓名+班级，每人限报 1 项

题目 1：FPS Demo 来源：天美

描述：基于虚幻 4 引擎的多人联机合作 PVE 的 FPS Demo，重点突出拟人的 AI 行为系统，如导航寻路、丰富的掩体行为、团队合作掩护战斗、群体 AI 行为等。

题目 2：TPS 闯关 Demo 来源：天美

描述：基于虚幻 4 引擎的 TPS 闯关 Demo， 重点突出 TPS 下角色 3C 操控(character, camera, controller)的丰富度和流畅爽快感，如滑铲、闪躲、攀爬翻越、滑索等，可以配合一些简单剧情闯关任务或闪躲攻击为主的简单战斗场。

题目 3：手机开发调试工具 来源：天美

描述：基于虚幻 4 引擎，制作更完善的手机开发调试工具，比如 Unity 的 Frame Debugger、代码 Hot Reloading、Shader、素材实时 Hot Reloading。

题目 4：脸动画编辑和驱动系统 来源：天美

描述：基于虚幻 4 引擎，和 Facial Action Coding System（facs），实现一个脸动画编辑和驱动系统。
基本目标：根据 facs 分解表情，形成表情元素。并制作编辑器方便使用者快速编辑面部动画和表情动画片段，并交付 UE4 的动画系统使用。
bonus： 结合语音识别、人工智能、机器学习领域的知识，自动将音频对话或者成段文字转化成面部动画。

题目 5：世界多同屏游戏后台云原生实践 来源：魔方

业务目标：以世界多同屏为技术背景，设计并实现一款游戏的核心玩法之后台系统。游戏前端可以竞技，也可以多人协作的开放世界、航海等，具体游戏体验自拟，不宜过重。
要求实现的功能至少包括：用户登陆、数据存储、地图（场景）管理、游戏玩法等。在时间可控范围内，可自行添加功能模块，以更好更全面的诠释真实的游戏运作。
技术目标：常见的游戏后端多是基于分布式系统构建，服务的运维和部署以往有一定难度和成本，kubernetes(k8s)很好地利用云上基础设施，提供了服务编排和管理。本课题要求以 k8s 为基础平台，将业务逻辑合理拆分，选择合适的协议与通信方案来构建。代码可托管在平台上(github/gitlib 等），通过自动化构建，可一键部署到云上并对外提供服务。希望方案无单点风险，在单台机器故障后不（尽量少）影响到游戏体验，并且核心服务性能高效满足尽量多的人同时可见。

核心技术点：如何搭建基础的游戏服务框架？怎么设计千人同屏玩法的后端及解决面临的性能挑战？ 如何运用云原生基础设施解决业务问题？ 工业化的流程管线怎么搭建？
项目收获：通过场景化的游戏核心逻辑的构建，了解游戏后台工作。在此过程中，学习容器与云原生技术，掌握基础架构能力和 devops 实践，并且较深刻地理解分布式系统的设计及测试相关理论知识。

题目 6：基于物理仿真的游戏实践 来源：魔方

业务目标 ：开发一个以物理仿真为主轴的游戏。一些例子：
1. 基于物理的迷题游戏玩法，如《Cut the Rope》、《Happy Glass》
2. 流体模拟，如《Where's My Water》
3. 非四轮载具模拟，如摩托车、坦克、直升机、机甲
4. 物理破坏，如地形 / 建筑爆破、刚体破碎、衣服 / 防具破裂
5. 物体燃烧传播、扑熄
6. 异常时间玩法，如《Braid》、《Prince of Persia: The Sands of Time》
7. 异常空间玩法，如《Portal》、《The Legend of Zelda: A Link to the Past》
游戏中可适量使用网上的美术资源。
技术目标：
1. 学习 Unity 或 UE4 游戏引擎的使用方法;2. 了解游戏中如何使用物理仿真方法；3. 把技术和创意游戏玩法结合；4. 可挑战自行实现物理仿真技术
项目收获：了解熟习游戏引擎、游戏开发过程，并针对物理仿真这题目尝试把技术与创意结合。

题目 7：真实感毛发模拟以及渲染 来源：光子

描述：在实时游戏领域毛发的表现一直都是一个比较棘手的问题，主要体现在毛发的物理模拟，碰撞检测，以及渲染对性能的消耗比较高，目前一些较为成熟的商业物理引擎也有毛发模拟解决方案，不过总体而言（主要是手机平台）其性能开销过大。本课题主要研究方向为：是否可以解决毛发物理模拟的性能问题（比如：利用 GPU 加速物理模拟，碰撞检测等），研究毛发的高质量渲染效果（毛发的透明部分处理，光照，自阴影，次表面散射等效果），最终能达到一个性能，效果都比较好的毛发效果，能在目前主流的手机游戏中应用

题目 8：游戏引擎移动端实时调试特性优化 来源：光子

描述：完善与优化 UE4/Unity 引擎移动端调试特性，提升 UE4/Unity 移动端特性开发效率，解决 UE4/Unity 移动端调试复杂、效率低的行业问题。通过该课题，可以全面了解 UE4/Unity 移动端游戏开发与调试，Android & iOS 系统 相关底层机制，移动端渲染管线等技术。

题目 9：游戏引擎客户端代码保护技术 来源：光子

描述：采用代码混淆、虚拟机改造等技术实现对游戏引擎的客户端代码进行定制化的增强保护功能。通过该课题，可以对游戏客户端安全、代码保护、混淆与虚拟机保护、调试与反调试、二进制逆向、Android & iOS 系统底层机制等技术有全面的了解。

题目 10：实时流体模拟 来源：光子

描述：实时流体模拟可以应用到游戏中的水流、烟雾、水流等效果，增强场景真实感。在 GPU 上利用一致性欧拉网格方法求解 shallow water 方程，解决模拟过程中存在的毛刺，斑块等问题，同时实现流体和刚体之间的交互模拟，最终可以整合到 UE4 中的粒子系统，体渲染等框架中，并实现出在手机平台上可运行的 demo。

题目 11：基于图的动作合成（Motion Sythesis based motion graph）
来源：光子

描述：准备一系列动画片段的动捕数据，通过一个 graph 模型驱动一个角色的 locomotion，能合成各个方向各个速度不同动作的合成和自然过渡。
一般算法思想：寻找动作数据中每帧的相似度，通过相似度来构建出过渡片段相连，从而组成一个 Graph，在运行时去检索这个 Graph 得到合适的动作。一般步骤如下：
1. Detection of Similarity
2. Generation of the Transitions
3. Graph Construction
4. Graph Search Meets the Goal
高阶目标：能识别复杂环境，不同斜坡，不同高度障碍等，作出合适的动作
开发环境：ue4，以一个动画蓝图的节点形式

题目 12：骨骼动画与环境的交互 来源：光子

描述：在 3D 交互式应用程序中，骨骼动画通过 3DMax、Maya 等工具制作并且在应用程序中播放，但常常出现固定的动画无法适应丰富的环境以至于表现效果生硬的现象，本课题将研究怎样在播放骨骼动画时更好的适应环境的问题。
课题要求：
1. 实现一个人形单位在较复杂的场景环境中自由移动的功能
2. 在移动的过程中动画够适配不同的速度、地面倾斜度，在遇到障碍物、沟渠时能用合理的表现自动翻越或跨越
3. 在移动过程中能够表现对一定方向范围内的目光注视
4. 能够在合理的场景或地形上进行攀爬，攀爬能适配不同的倾斜度
5. 能够用合理的动作拾起身边小范围内的任意物件
6. 在任何情况下，与环境发生接触的肢体不能有和接触点之间的不合理的滑动（或尽量减少）
7. 在任何情况下，不能有不合理的抓空、踏空等现象（或尽量减少）
收获：能够了解当下前沿的骨骼动画

题目 13：高真实感复杂光照环境场景模拟 来源：光子

描述：基于 UE4 引擎，表现一个高度真实的带有复杂光源环境的场景。该 demo 需要能够流畅运行于主流的移动设备上（如 android 或 ios 手机）。
场景中至少要表现或支持以下元素，在此之上还可随意发挥:

1. 任意数量的动态光源照射（一盏带阴影的平行光，多盏点光和聚光灯）
2. 海水表现（海浪，海水和海岸交互，海水反射，折射等）
3. 天空表现（大气散射，云，太阳等）
4. 天气变化（天空，光照，降水等）
5. 风的表现及与植被(树木和草）的交互
6. 可以动态切换画面的质量等级以适应不同的设备
7. 可以控制角色或摄像机在场景中漫游

收获：1. 使学生对当今游戏渲染的前沿技术（延迟渲染，自然环境模拟等）进行探 索和实现，并有可能将成果运用到具体项目中。
2. 课题的发挥余量比较大，可激发学生的创造力，尽可能表现更真实的自然环境。

题目 14：高性能 UI 框架解决方案 来源：光子

描述：目前主流商业引擎 Unreal Engine4 和 Unity，他们的 UI 系统在渲 染绘制流程中，cpu 和 gpu 的耗时都非常高，一直都是各个项目组的性能痛点。这个方案旨在解决 UI 系统的性能问题，主要是想实现并解决下面几点：

1. 不需要无谓的 UI 层级树的遍历检测 （ cpu friendly）
2. 层级树设计结构带来的大量 Object 数量 （memory friendly）
3. 精简状态管理，实现局部状态更新机制，消灭父子节点递归更新（cpu friendly）
4. no state changes（shader，blend…）（cpu，gpu friendly）
5. one draw call （cpu，gpu friendly）
6. GPU Soft clipping（cpu friendly）
7. GPU Vertex transform （ cpu friendly）
8. Less overdraw （gpu friendly）

题目 15：分布式系统一致性算法课题研究 来源：光子

描述：分布式系统通过大规模的服务器集群协作提供大规模的计算服务，以其巨大的成本优势、计算弹性、整体高可靠性等优势，在工业应用领域已经全面替代了中央大型机的模式。然而，在数据存储和强状态的服务的场景中，分布式系统中的计算节点故障切换过程中，会导致数据错误或者丢失的问题。当一个缓存的最新数据的进程故障时，倘若切换到另一个替代进程，虽然可以快速恢复服务，但是故障的进程上所缓存的其它节点并不知道的数据会彻底丢失。更困难的是，在分布式系统中，没有一个可靠的方法判断一个进程是故障了还是因为网络分区导致该进程暂时不可通信。因此，在大多数情况下，都不得不根据业务场景进行有选择的牺牲。牺牲系统可用性或者牺牲数据一致性。

分布式一致性算法 （Paxos，Raft 等）的出现提出了一个新的问题解决的方向，通过多个实例进行相同指令序列的计算，少数节点故障或者不可通信时，同时保障可用性和数据一致性。Google 的 MegaStore, Spanner, Chubby 等系统，均基于 Paxos 算法实现。国内近些年，也逐步在工业领域里进行应用。比如腾讯的 PaxosStore。 已有的工业应用实践，都还有改良改善的空间。有的系统仍旧需要十几秒的时间才能恢复服务，有的系统无法支持并发写入。性能和成本也是一个巨大的挑战。

题目 16：分布式图计算与可视化 来源：光子

任务 1：海量游戏图数据计算
描述： 在大规模用户群体的游戏中，利用多种维度的社交互动(例如点赞、组队、送礼等)和好友关系链网络进行如下计算和分析数据流搭建：从日志生成，到数据流缓存再到图数据端，精准构建网络并尽可能缩短图网络构建和刷新延迟。

预期目标：
- 图规模： 亿级别节点，百亿级别边，包含 5 种以上社交互动以及部分节点特征
- 数据构建：利用分布式数据流快速构建和更新图网络，并支持后续高并发的查询

测试数据集： 3 亿节点，40 亿条边
运行时间： 4h
社区规模： 总社区数达几千万左右
任务价值：挖掘和分析当前游戏的用户体验、优化游戏内社交互动环境、提升玩家留存与活跃。

任务 2：海量数据可视化
描述：探索合适的展示形式，浓缩在海量对局中用户坐标、状态等特征，输出以地图为底图展示用户流向、状态等可视化动图。
目标：在 80 万*80 万的坐标系地图上，展示亿级别用户，在不同时间的坐标、发生战斗记录、死亡情况提供算法和可视化组件，在输入相关数据后、在可用时间内输出可视化展示图产出一个可使用的可视化组件。
任务价值：观察分析用户的行为情况以及游戏节奏状况，辅助提升游戏体验。

题目 17：Web Assembly 技术开发 H5 游戏 来源：光子

描述：用 UE/Unity 采用 Web Assembly 技术开发 H5 游戏，让 H5 游戏能够运行在浏览器、手 Q、微信等环境中，并进行性能上的调优，包括 H5 包的大小、对运行环境的内存、CPU、 GPU 资源的要求、网络数据等，确保游戏的体验达到手游，PC 游戏的水平。

题目 18：Runtime Procedural Destruction 来源：光子

描述：运行时的实时破坏：实时生成随机形状，对模型进行切割、更新渲染和物理，达到真实物理破坏效果。通过这个课程可对 UE4 中模型数据、渲染流程、物理等有较深入了解。

题目 19：Mesh Toolkit 课题 来源：光子

描述：Mesh Toolkit 是一个处理 mesh 工具集，可以帮助 Artist 解决 mesh 制作不符合规范的地方，也可以在 Artist 制作的 mesh 基础上生成简化模型用于提高性能，或者生成供其他模块使用的数据。主要功能如下：
1. 在现有 mesh 的基础上，重新生成三角面更合理的 mesh；

2. 删除 mesh 中看不见的三角面；
3. 根据要求切分 mesh；
4. 在现有 mesh 的基础上，遵循最小遮挡原则，生成极简化面片，这些面片用于在游戏场景中做为遮挡体做视野遮挡。（可以参考：https://arisilvennoinen.github.io/Publications/Occluder_Simplification_using_Planar_Sections_EGSR_Notes.pdf）
5. 在现有建筑 mesh 的基础上生成 portal。目的是找到建筑的开口，如门、窗户，用于区分室内不同区域以及室外；
6. mesh 简化模型的自动生成。可以根据 artist 指定区域，保留形态外观的前提下，减少面数；
7. 自动生成 mesh 的 UV 坐标，做为 mesh lightmap 的 UV，确保三角面在纹理上的映射不会相交而且连续。

题目 20：Advanced GPU Baking for Unreal Engine 来源：光子

描述：研究 GPU 光线追踪在游戏渲染领域的应用，在 GPU 架构上实现光线追踪相关渲染算法，采用最新的渲染技术对其进行优化，改善传统的光照贴图以及光照探针等全局光照解决方案的预计算处理流程。进一步探索辐照度缓存等高级算法在光照烘焙领域的应用，调研新的光线追踪降噪技术以改善渲染烘焙结果。将 GPU 烘焙系统同 Unreal Engine 进行集成，实现工业级的 GPU 分布式 光照烘焙。

本课题涵盖离线渲染的基础知识，光线求交加速结构，BSDF 材质计算，光源模型，高级路径追踪算法，随机采样技术，降噪技术，3D 场景信息描述，全局光照以及相关工程优化技术。

预备知识：微积分，线性代数，概率论，数据结构与算法分析，计算机图形学基础，C/C++算