南京师范大学

毕业设计（论文）开题报告

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名：** | 朱正阳 | **学 号：** | 19210217 |
| **学 院：** | 计算机与电子信息学院/人工智能学院 | | |
| **专 业：** | 计算机科学与技术 | | |
| **题 目：** | VR APP的自动化异常检测工具  设计与实现 | | |
| **指导教师：** | 段博佳 | | |

2025年1月3日

开题报告填写要求

1．开题报告作为毕业设计（论文）答辩委员会对学生答辩资格审查的依据材料之一。此报告应在指导教师指导下，由学生在毕业设计（论文）工作前期内完成，经指导教师签署意见及院、系审查后生效；

2．开题报告内容必须用黑墨水笔工整书写或按教务处统一设计的电子文档标准格式（可从教务处网址上下载）打印，禁止打印在其它纸上后剪贴，完成后应及时交给指导教师签署意见；

3．有关年月日等日期的填写，应当按照国标GB/T 7408—94《数据元和交换格式、信息交换、日期和时间表示法》规定的要求，一律用阿拉伯数字书写。如“2024年1月10日”或“2024-01-10”。

**毕 业 设 计（论 文）开 题 报 告**

|  |
| --- |
| **1. 本课题的目的及研究意义** |
| **研究目的：**  近年来，随着元宇宙（Metaverse）概念的兴起以及智能头戴设备（Head-Mounted Display，HMD）的快速发展，虚拟现实（Virtual Reality，VR）技术逐渐走进大众视野。虚拟现实通过计算机模拟具有视觉、听觉和触觉效果的人造环境，旨在为用户提供高度沉浸式的体验。VR应用（VR APP）主要指运行在Oculus、Pico等智能头戴设备上的应用程序。当前主流HMD的操作系统通常基于安卓（Android），且大部分应用采用Unity引擎进行开发，这使得VR应用不仅面临传统Android应用的常见问题，还遭遇与Unity引擎特性相关的额外挑战。  在本课题中，VR应用中的异常主要来源于开发过程中的不当代码实践、不兼容的Unity及其插件包版本等因素。这些问题通常表现为两类：性能问题（Performance Bug）和功能性问题（Functional Bug）。性能问题通常由不当的代码实践引起，如堆分配过度导致GC压力过大，进而影响CPU性能；或图形模型的边数过多，导致GPU负担加重，造成帧率不稳定；资源模型错误可能导致画面出现UI闪烁（如紫色错误）。功能性问题则主要包括软件崩溃（Crash）、空指针引用（Null Reference）等错误，严重影响应用的稳定性和用户体验。  此外，VR应用的项目文件数量庞大，场景和模型资源复杂多样，给VR应用中的异常检测带来了极大的挑战。目前，主流的VR应用公司和游戏公司通常依赖大量的黑盒和白盒测试来发现和报告程序缺陷，这不仅消耗了大量人力资源，而且测试覆盖面有限。虽然已有一些框架试图解决VR应用中的自动化异常检测问题，例如VRTest和VRGuide，但它们在场景探索（Scene Exploration）中仅通过简单点击操作进行检测，未能模拟抓取、拖拽等更复杂交互的场景检测。iv4XR则通过交互序列化建模进行场景探索，但仍依赖简单的有限状态机（Finite-state machine，FSM）模型，无法全面涵盖VR应用的交互复杂性。  针对以上问题，本课题从软件工程和软件可靠性的角度出发，重点研究如何自动化检测Unity引擎开发的VR应用中的常见异常。通过引入自动化场景探索技术，全面触发场景中的交互事件，并结合软件工程方法和静态分析手段，以提升VR应用开发的质量与效率，减少手动测试所带来的时间和成本。  本课题的预期贡献如下：   1. 设计并实现一款综合性的自动化异常检测工具。该工具预期会通过基于传统的寻路算法、大语言模型（Large Language Model，LLM）算法和强化学习（Reinforcement Learning, RL）算法，核心任务是场景自动化探索，全面触发VR场景中的交互事件，通过动态检测与静态分析相结合的方式，帮助开发者及时发现并消除代码中的不当实践与异常问题，从而显著提升VR应用的开发质量与用户体验。 2. 构建Unity VR项目数据集。通过从GitHub/GitLab收集开源的Unity VR项目，并进行人工二次验证，构建一个用于交互和场景检测的VR应用数据集。这一数据集将为自动化异常检测工具提供实际的测试数据，进一步验证工具的检测能力与实用性。 3. 评估与对比工具的效果。通过设计合理的评估指标，并在构建的VR项目数据集上进行验证，评估该工具的异常检测能力，与现有的相关方法进行对比，从而为VR应用开发中的自动化检测提供有力的实践依据。   **研究意义：**  根据《财富》报道，2023年，全球VR市场规模为251.1亿美元，预计将从2024年的326.4亿美元增长到2032年的2448.4亿美元[1]。这说明随着VR技术发展，VR的市场增长迅速。VR设备（包括控制器，HMD，传感器等）变得更加复杂、精细和现代化。  VR技术的应用也渗透到人们各行各业，在医疗、教育、航空、军事、工业生产领域和日常娱乐等方面有着广泛应用。首先，VR技术在教育领域的应用逐渐受到重视。通过虚拟环境，学生能够进行互动式学习，进行实验、模拟和角色扮演等，这对于传统的教学模式是一种极大的补充。其次，在医疗领域，VR的应用则主要集中在手术模拟、病患康复等方面，医生可以在虚拟环境中进行手术演练，减少实际手术的风险，而病患也可以通过虚拟场景进行康复训练，达到更好的治疗效果。VR在娱乐行业中的应用也很广泛，尤其是在游戏领域。通过VR设备，用户可以进入一个完全虚拟的世界，体验极致的游戏互动。这种高度沉浸感的体验，使得传统的2D或3D游戏无法比拟[2]。  除此之外，VR还在建筑设计、房地产展示、社交娱乐、心理治疗、远程工作等多个领域展现出巨大的潜力。随着元宇宙的构建和5G技术的普及，VR技术的应用场景将进一步扩展，成为数字世界与现实世界之间的重要桥梁。  然而，随着VR技术日益普及，VR应用在开发和使用过程中所面临的技术挑战也逐渐浮现，尤其是在使用Unity引擎开发的VR应用中，这些问题的存在更为突出。  VR应用作为一种高度复杂的软件系统，其在开发过程中容易产生多种异常，主要包括性能问题和功能性问题。性能问题和功能性问题是影响VR应用稳定性和用户体验的两大类重要问题，它们各自源于不同的开发环节，具有不同的特点和挑战[3]。  性能问题是指由于不当的代码实践或不合理的资源管理，导致VR应用在运行过程中出现响应迟缓、帧率下降等问题。VR应用尤其需要保持较高的帧率（通常要求至少30-60帧/秒）以确保用户体验的流畅性，若帧率过低，容易导致VR头显中的运动伪影和延迟，严重时可能导致晕动症等不适症状。常见的性能问题通常来源于以下几个方面：  垃圾回收压力过大：在Unity引擎中，垃圾回收（Garbage Collection，GC）是一种常见的性能瓶颈。频繁的堆内存分配和不当的内存管理可能导致垃圾回收压力过大，进而影响CPU的性能。尤其是在VR应用中，长时间运行可能导致GC频繁触发，进而导致帧率下降。  过于复杂的图形模型：VR应用的图形表现往往要求较高的渲染质量，而复杂的三维模型和大量的多边形会增加GPU负担，影响渲染效率。尤其是一些具有大量多边形的模型，可能会导致图形渲染的瓶颈，造成帧率波动或画面卡顿。  资源模型错误：VR应用中的资源管理至关重要，如果模型资源没有得到有效优化，可能会导致不必要的资源浪费，甚至影响UI的渲染效果。例如，当使用错误的纹理格式或未优化的贴图时，可能导致UI闪烁或画面出现紫色错误等问题。  功能性问题主要是指软件由于设计缺陷或代码实现错误，导致应用的功能无法正常工作，或产生不符合预期的行为。常见的功能性问题包括：  软件崩溃（Crash）：当程序在运行过程中遇到未处理的异常或致命错误时，可能导致应用崩溃。在VR应用中，崩溃不仅会影响用户体验，还可能导致用户的设备卡死或重启，造成很大的不便。  空指针引用（Null Reference）：空指针错误是开发中最常见的一类错误，尤其是在Unity中，由于对象生命周期管理不当，可能会发生空指针引用，从而导致应用崩溃或功能异常。  内存泄漏：内存泄漏是指程序分配了内存，但没有及时释放，导致系统内存占用逐渐增加。长时间的内存泄漏会导致应用性能下降，甚至系统崩溃。  这些问题虽然看似与性能无关，但它们直接影响VR应用的稳定性，进而影响用户的整体体验。因此，在VR应用的开发过程中，必须时刻关注这些潜在的功能性问题。  VR应用的开发过程中，异常和漏洞的检测一直是一个难题。首先，VR应用涉及的场景和模型非常复杂，尤其是在大规模的场景和多重交互中，开发者很难在有限的时间内覆盖到所有可能的交互情境。其次，VR应用的测试不仅仅是功能测试，还包括对性能的检测，这对硬件和软件的协同能力提出了更高的要求。传统的测试方法，如黑盒和白盒测试，往往依赖人工进行大量的测试，这种方法虽然能够发现一定数量的问题，但由于人力资源的限制，难以覆盖所有的场景和交互。  自动化测试作为一种解决方案，虽然可以通过编写脚本模拟用户操作，覆盖更多的场景，但传统的自动化测试方法通常依赖简单的操作模拟，无法有效捕捉到复杂的交互行为。此外，现有的自动化检测工具大多无法模拟VR特有的交互方式，如抓取、拖拽等，这使得它们在VR应用的检测中无法发挥出应有的作用。  针对VR应用中的异常检测，已有一些相关工作尝试提供自动化解决方案。例如，VRTest和VRGuide通过模拟点击操作进行自动化测试，但这些方法仅能覆盖简单的用户交互，如按钮点击等。它们对于复杂的交互行为，例如对象抓取、物理碰撞等，无法提供有效的支持。  iv4XR通过交互序列化建模的方法尝试对VR应用中的交互行为进行建模，从而进行自动化的场景探索。尽管这种方法在一定程度上提高了检测的覆盖率，但由于其依赖简单的有限状态机模型，仍然无法全面应对VR应用中复杂的交互情境。有限状态机模型无法处理复杂的交互序列，也无法应对动态变化的场景。  本课题开发的框架基于智能体进行场景探索。在智能体的决策、动作执行和寻路问题中，拟应用诸如LLM、强化学习、NavMeshAgent、Git等工具。  LLM是一种基于深度学习技术的自然语言处理（Natural Language Processing，NLP）模型，通常由数十亿甚至千亿级别的参数组成，能够对自然语言文本进行理解、生成和推理。这类模型通过在大规模语料库（例如文本数据集、代码数据集）上进行训练，具备强大的语言建模能力，可用于文本生成、问题回答、代码补全和语义理解等任务。在场景探索过程中，LLM可以用于对任务（Task）和动作（Action）进行序列化建模。通过学习语义关联，生成合理的任务执行顺序（例如“抓取-投掷-触发按钮”），帮助智能体更高效地完成目标。  强化学习是一种机器学习方法，通过与环境的交互来学习一种能够最大化长期回报（Reward）的策略。强化学习的核心思想是，智能体通过观察环境状态（State），选择一个动作（Action），从环境中获得反馈（Reward），并更新策略（Policy）以优化未来的决策。在VR场景中，，智能体需要探索复杂的环境以触发潜在的程序漏洞。强化学习能够帮助，智能体学习一种基于长期回报的策略，使其在复杂场景中更高效地选择最优路径和动作。  Git是一种分布式版本控制系统，广泛应用于代码的版本管理。Commit指的是将代码修改提交到Git仓库的一个操作，每次提交都记录了代码的变更内容、提交时间和作者信息。Commit记录是代码仓库的重要组成部分，通过这些记录可以追溯代码的历史变更，分析项目在不同时间点的状态。为了验证VRAgent框架的有效性，需要将其触发的程序漏洞与实际存在的程序漏洞进行对比。通过分析Git仓库的Commit历史，能够找到开发者提交的程序漏洞修复记录，作为程序漏洞的“Ground Truth”（真实数据）。  本课题提出的研究方法论结合了多种先进技术，目标是为VR应用开发中的异常检测提供一种高效且全面的自动化解决方案。该方法论的核心是基于智能体的场景探索框架，整合了强化学习、LLM以及Git工具等技术，为VR应用中的异常检测提供了新的视角和创新的技术路径。在VR技术日益广泛应用的背景下，解决异常检测问题具有重要的现实意义和潜在的市场价值，能够为VR行业的稳定发展、提高用户体验和推动技术进步做出积极贡献。 |
| **2．本课题的国内外的研究现状** |
| 现有的VR异常实证研究涵盖了从性能优化到隐私安全的多个方面，为异常检测工具的开发提供了丰富的经验和方法；在场景探索和寻路方面，现有的工作提及到的交互序列问题和复杂的交互动作问题仍然比较难以解决；在VR智能体的研究方面，现有的工作已经在游戏检测领域初步尝试结合了LLM和强化学习方法。  **2.1 VR 应用程序漏洞实证研究**  VR应用程序漏洞的实证研究，主要是指通过对开源VR项目的程序漏洞的统计性分析，提取其关键特征，并进行分类，以及进一步的分析每一种漏洞的产生原因。这是漏洞检测工作的关键前置问题，只有明确漏洞及其产生类型才能更好的去检测漏洞、评估工具的性能。目前国内外的研究中，软件工程领域有一些针对游戏漏洞和VR应用漏洞的实证分析，其中的分析方法很值得借鉴。  文献[4] 是网易雷火事业部伏羲AI Lab联合德克萨斯大学达拉斯分校关于对游戏中的画面异常（UI Glitches）问题进行了研究。文章首先阐明了8种画面异常问题和4大类产生问题的代码层级的原因，然后通过代码注入，增强画面异常截图数据集，通过CNN训练队画面异常截图分辨模型，通过梯度得到画面异常显著图，帮助程序员定位画面异常。  该方法的借鉴意义在于对异常特征分析以及代码注入增强数据集的方法。由于其是关注画面截图的表现，更多倾向于计算机视觉（Computer Vision，CV）的角度去识别画面异常；并且该方法更多的是关注于传统游戏，而并没有讨论VR应用。  文献[5] 是德克萨斯大学圣安东尼分校联合上海交通大学针对VR应用开发者如何优化性能问题的实证研究（Empirical Study）。作者通过对45个开源VR项目数据集的git commits messages 手动分析（两位研究者独立分析，合并、讨论确定最终结果，通过Cohen's-Kappa值达成一致）；然后通过静态分析（Static Analysis），使用srcML解析C#代码，GumTree解析抽象语法树，通过MetaID检测unity依赖文件，将性能问题的优化方法分为图形简化、渲染优化、API代码优化、堆分配避免和值捕获。最终说明了优化方法在官方VR项目和独立开发者之间的区别等若干研究性问题。  该论文对VR 性能问题做了深刻的特征分析，并且从代码角度说明了性能问题产生的原因和解决方法。  文献[3] 是中山大学软件学院和香港浸会大学合作的提出了一款名为VR-SP Detector的工具，用于检测主要面向Unity Oculus VR应用的安全性和隐私性漏洞的工具。首先通过对apk包合政策隐私协议的提取，通过PolicyLint进行分析比对隐私方面的漏洞；然后对manifest文件进行分析，并通过污点追踪，最终得出结论，隐私和安全性问题在VR应用中很常见。  该论文主要关注隐私和安全性漏洞，重点在于对打包后的apk文件以及隐私政策文本的分析，以及逆向工程后对源代码的分析。本课题的VRAgent是基于源代码的分析，可以借鉴文章中提到的对C#代码的静态分析方法、以及对git commit messages的分析方法。  **2.2 VR和寻路算法和场景探索**  寻路问题是一个非常经典且被充分研究的问题，在计算几何和图论领域中，经典的最短路算法比如Dijkstra和Flody；游戏中的寻路算法比如A\*算法等；全局优化问题中的Prim和Kruskal算法用于解决最小生成树等。在复杂的VR和3D游戏中，寻路问题需要考虑更多。例如每帧更新的动态的障碍物，以及代价问题（最短路可能会有其他游戏代价）和智能体的随机性问题（比如偶尔走入“危险区域”更符合拟真玩家的行为）。而场景探索，则是VR测试中需要解决的问题。在寻路的基础之上，VR测试智能体需要尽可能的去探索场景，而除了障碍物以外，还有可能出现交互之后才能继续探索的区域、事件等。这给场景探索带来了复杂性和多样性，简单的DFS等暴力方法无法解决。  文献[6] 探讨3D游戏中的寻路算法问题，包括动态避障和精细控制（比如有选择性的走入危险区域），也提到了决策序列问题。  该论文发现Unity中的插件NavMesh可以很好的解决寻路问题。这是一个底层基于启发式的A\*算法和图形学方法实现的寻路插件。通过对场景模型的Mesh构成的图进行Dijkstra等最短路算法的启发式综合进行最短路求解，同时能够很好的解决动态避障问题。操作上只需要通过烘焙静态网格以及设置NavMeshAgent即可进行寻路。  文献[7] 是VR场景探索的一篇相关工作。VRTest是德克萨斯大学圣安东尼奥分校Xiaoyin Wang老师提出的场景自动化探索工具，通过自动控制相机移动、追踪物体交互事件（点击等），探索可交互物体。该方法首先计算拥有renderer组件，即可见物体的Bounding Boxes包围盒；然后通过状态变化报告器，获得物体的 EventTrigger组件，获得组件的Entry条目，对回调增加监听，将状态变化报告器添加到所有可交互物体，在对应event触发时向上报告。并且通过Evaluation实验，对比了Moneky随机算法和Greedy贪心寻路算法，通过若干指标：可交互物体的探索程度、时间效率等，对5个开源VR项目进行实验，并检测异常。  文献[8]是VRTest的改进工作，主要通过计算几何方法，对VRTest的相机寻路进行了优化。文章提到了Art Gallery Problem 美术馆问题和Watchman Route Problem 观察者路由问题 。这是一个关于在多边形美术馆里，选择保安的最短路径，让他的沿途可以看到每一个角落的问题。传统的方法是The Cut Theory。而VRGuide提出了动态割寻路，计算距离最近的割的距离；相比VRTest的Greedy寻路，得到提升。在评估时，该工作将VRTest作为Baseline，将测试案例扩展到8个VR应用。  这两篇工作的核心都是场景探索。Unity中有复杂的游戏场所，让开发人员手动探索是非常耗时耗力的。这两篇工作初步将模拟交互的思想放到VR测试中。然而，目前的交互仅限于模拟“鼠标点击”这一简单操作，无论是VRTest还是VRGuide都没有考虑到复杂的VR交互，比如说抓取物体、开门、拉弓箭等操作；同时也无法处理需要一定固定交互序列的操作，比如先找到钥匙，再开门然后再拉下电闸。因此，本课题希望能够处理更复杂的VR交互，并且能够通过强化学习和LLM的方法，提高场景探索的能力。  **2.3 VR应用测试和智能体**  软件测试是软件生命周期中的重要组成部分。VR应用作为一种软件，需要严格和规范化的自动测试，更需要自动化的测试工作流。目前国内外提出了基于代码异味的检测、基于强化学习算法的智能体检测等。其中，智能体检测是指通过一个模拟玩家的智能体在场景中漫游，去运行游戏中的所有分支情况，尽可能的去触发函数执行链条，从而达到检测功能性问题的目标。  文献[9]提出了VR 应用可用性自动化测试方法，User Study 验证可行性（用户可以自行探索场景，自动记录任务树，可用性异味检测）。扫描场景中的所有可交互VR物体，找到其中具有event handling脚本的，并扩展使其能够在event log的时候，自动化用户行为记录（保存到log file）：物体抓取、释放、物体使用、物体不使用、头部移动；将这些记录保存到中央服务器；提出了任务树生成算法，任务树能够代表记录的用户行为。  该论文关注VR应用的可用性测试，从用户的角度出发，需要用户自己进行场景探索的同时，框架在后台记录Task Tree和行为日志，通过对Task Tree的规律分析，寻找VR应用的可用性异味（Smells）。然而该框架仍然需要用户去自己探索场景，并不能达到自动化场景探索和异常检测的目标。  文献[10]是网易雷火事业群伏羲AI Lab提出的一款基于演化强化学习的针对多人在线对战游戏的测试智能体。Wuji模型忽略了GUI，简化为考虑游戏技能释放和移动方向；首先对异常特征进行分析，对异常分成了5类；评估时通过手动注入异常作为“Groud Truth”，分析了两款网易游戏；衡量指标采用代码覆盖率、状态覆盖率。  该论文创新性的将强化学习引入了游戏测试智能体。然而，Wuji无法解决VR应用中的场景探索问题，因为它只能学会释放技能和移动，并不具备处理交互物体的复杂逻辑。  文献[11]提出了名为iv4xr的游戏检测智能体。通过对游戏任务和action进行序列化建模，不关心底层实现细节，而是在高层上抽象任务结构（Goal Struction）和行为（Action），需要程序员手动去设定序列；通过接口和底层函数（如寻路）交互。  该方法创新性的将交互动作进行序列化建模。然而这一工作需要程序员手动进行。因此如果能够通过LLM对Unity场景和代码文件进行学习，自动生成结构化、形式化模型，也许能够更好的解决智能体的决策问题。  文献[12]这篇工作针对VR应用开发测试做了实证性研究（测试代码比重、测试有效性），主要阐明了VR应用在开发过程中的自动化测试方法和评估测试样例的性能，并对测试样例进行了分类。  该论文考虑到了传统安卓应用和VR应用之间的差异性问题，比如考虑到了VR应用的交互测试、物理碰撞测试等。在方法构成上有借鉴意义。 |
|  |

|  |
| --- |
| **3. 本课题的研究内容** |
| 课题的研究内容围绕构建高效的自动化检测框架，主要包括以下几个方面：  首先，开源VR应用数据集构建是研究的起点。通过对Github和Gitlab等开源平台上的Unity VR相关项目进行爬虫抓取与人工筛选，希望构建一个高质量VR项目的数据集数据集中的项目提供了大量的真实VR应用实例，包含了丰富的VR场景和模型，可以为后续的异常检测与分析提供实际的测试环境。  其次，VR Agent场景探索智能体是本课题的核心。通过结合NavMesh技术，解决了VR应用中的寻路与动态避障问题，并采用回溯与分支界限优化算法进行目标选择与寻路优化。此外，课题还希望引入LLM技术，通过学习场景信息和源代码，完成任务和动作的序列化建模。尽可能探索是否可以结合强化学习和有限状态机控制策略，力求让VRAgent能够智能地选择和执行动作，从而在复杂的VR场景中进行有效的异常触发与异常检测。  其中，关于强化学习的研究将成为一个难点。目前在Unity中的MLAgent插件下，能够实现基于强化学习网络的能完成简单移动的智能体。下一步的工作是将多种交互加入到强化学习中，尝试让智能体学会多种交互。  而LLM构建交互序列模型也将是难点之一。目前LLM针对UnityVR源码的模型并没有很多，如何训练和评估模型将是重难点。  最后，框架可用性验证则通过在构建的数据集上进行VRAgent测试，评估该框架的实际效果。课题尽可能对比其他异常检测方法，并使用代码覆盖率、可交互物体覆盖率、物理交互覆盖率等多个指标来评估框架的性能。  通过以上内容，课题的研究不仅能提供一种创新的自动化异常检测方法，也为VR应用开发中的异常检测提供实用的工具和方法，具有重要的学术和实际应用价值。 |
| **4. 本课题的实行方案、进度及预期效果** |
| **实行方案**：   1. 开源VR应用数据集构建工作 构建一个Unity VR应用数据集，收集运行时异常，并分析关键特征，对VR 运行时异常进行分类和代码级别的原因分析。 具体收集方法：通过人工和爬虫方法，在Github/Gitlab等开源仓库上收集有关'unity', 'unity-vr', 'vr'的开源仓库，通过比较Packages/manifest.json 清单文件，分析和VR应用相关性较高的开源仓库作为初筛数据集。   接着通过人工验证项目的原创性、丰富程度（比如是否包含足够的VR场景和模型）、Lib库兼容性（比如是否出现项目过旧，其插件或者类已经无法使用），进行项目复筛。 目前从Github收集971个仓库，经过自动和手动两轮筛选保留了104个质量比较高的VR项目。   1. VRAgent场景探索智能体   核心在于VR应用的场景探索。   1. 通过使用NavMesh解决寻路中的动态避障和代价问题； 2. 关于多个可交互物体的执行顺序决策，加入基于回溯和分支界限的全局优化算法进行目标选择算法和寻路上的优化； 3. 尽可能通过LLM方法对场景信息和源代码进行学习，对任务和行为进行序列化建模，基于强化学习的方法和基于有限状态机的方法控制智能体的策略选择和动作执行，从而进行场景探索和异常触发。 4. 框架可用性验证   将VRAgent在数据集上进行测试，尝试触发异常；通过Commit历史上的fix版本回溯，作为异常的“Ground Truth”，并且对比其他方法，分析代码覆盖率、可交互物体覆盖率、物理交互覆盖率等指标作为评价，验证框架性能。  **进度安排：**  2024.10.22-2024.11.16：查阅文献资料，并进行整理分析，确定选题。  2024.11.17-2024.12.31：阅读国内外LLM、强化学习、场景探索、游戏寻路、应用测试方面的论文，学习并掌握基础算法，尝试复刻已有框架。  2025.1.1-2025.3.31：重点设计和实现VRAgent的各项功能。  2025.4.1-2025.4.30：评估实验结果，进行分析，撰写论文，并交由导师审阅，完善论文。  2025.5.1-2025.5.15：完善论文，准备答辩材料。  **预期效果：**   1. 数据集构建方面，最终能够保持数量在100个以上规模的高质量开源VR项目数据集。 2. 实现VR Agent的寻路模块、决策模块、场景探索模块。尽可能的融入LLM算法和强化学习算法，投稿智能体的决策能力和交互能力。 3. 评估模型，尽量提高模型的异常检测能力到80%以上，并且希望能够发现两位数以上的未发现的异常；对比Baseline模型能够有很好的效率和覆盖率； |
|  |

|  |
| --- |
| 5. 已查阅参考文献 |
| 1. F. B. Insights,‘Virtual reality (vr) market size, share & industry analysis, by component (hardware, software, and content), by device type (head mounted display (hmd), vr simulator, vr glasses, treadmills & haptic gloves, and others), by industry (gaming, entertainment, automotive, retail, healthcare, education, aerospace & defense, manufacturing, and others),and regional forecast, 2024-2032,’ https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-market-101378, 2024, accessed: 2024-09-04 2. T. Mazuryk and M. Gervautz,’Virtual reality - history, applications, technology and future’, Institute of Computer Graphics, Vienna University of Technology, Tech. Rep., Dec 1999, article. 3. H. Guo, H. -N. Dai, X. Luo, Z. Zheng, G. Xu and F. He, "An Empirical Study on Oculus Virtual Reality Applications: Security and Privacy Perspectives," 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE), Lisbon, Portugal, 2024, pp. 1958-1970, doi: 10.1145/3597503.3639082. 4. Ke Chen, Yufei Li, Yingfeng Chen, Changjie Fan, Zhipeng Hu, and Wei Yang. 2021. GLIB: towards automated test oracle for graphically-rich applications. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2021). Association for Computing Machinery, New York, NY, USA, 1093–1104. https://doi.org/10.1145/3468264.3468586 5. F. Nusrat, F. Hassan, H. Zhong and X. Wang, "How Developers Optimize Virtual Reality Applications: A Study of Optimization Commits in Open Source Unity Projects," 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), Madrid, ES, 2021, pp. 473-485, doi: 10.1109/ICSE43902.2021.00052. keywords: {Rendering (computer graphics);Animation;Software;Real-time systems;Complexity theory;Optimization;Software engineering;Empirical Study;Virtual Reality;Performance Optimization}, 6. I. S. W. B. Prasetya, Maurin Voshol, Tom Tanis, Adam Smits, Bram Smit, Jacco van Mourik, Menno Klunder, Frank Hoogmoed, Stijn Hinlopen, August van Casteren, Jesse van de Berg, Naraenda G.W.Y. Prasetya, Samira Shirzadehhajimahmood, and Saba Gholizadeh Ansari. 2020. Navigation and exploration in 3D-game automated play testing. In Proceedings of the 11th ACM SIGSOFT International Workshop on Automating TEST Case Design, Selection, and Evaluation (A-TEST 2020). Association for Computing Machinery, New York, NY, USA, 3–9. https://doi.org/10.1145/3412452.3423570 7. X. Wang, "VRTest: An Extensible Framework for Automatic Testing of Virtual Reality Scenes," 2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Pittsburgh, PA, USA, 2022, pp. 232-236, doi: 10.1145/3510454.3516870. keywords: {Greedy algorithms;Training;Quality assurance;Video on demand;Virtual reality;Cameras;Software;Software Testing;Virtual Reality;Scene Exploration}, 8. X. Wang, T. Rafi and N. Meng, "VRGuide: Efficient Testing of Virtual Reality Scenes via Dynamic Cut Coverage," 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE), Luxembourg, Luxembourg, 2023, pp. 951-962, doi: 10.1109/ASE56229.2023.00197. keywords: {Geometry;Quality assurance;Computer bugs;Virtual reality;Cameras;Software;Testing;Software Testing;Virtual Reality;Scene Exploration}, 9. Patrick Harms. 2019. Automated Usability Evaluation of Virtual Reality Applications. ACM Trans. Comput.-Hum. Interact. 26, 3, Article 14 (June 2019), 36 pages. https://doi.org/10.1145/3301423 10. Y. Zheng et al., "Wuji: Automatic Online Combat Game Testing Using Evolutionary Deep Reinforcement Learning," 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), San Diego, CA, USA, 2019, pp. 772-784, doi: 10.1109/ASE.2019.00077. keywords: {Games;Computer bugs;Testing;Space exploration;Reinforcement learning;Manuals;Sociology;Game Testing;Artificial Intelligence;Deep Reinforcement Learning;Evolutionary Multi-Objective Optimization}, 11. Samira Shirzadehhajimahmood, I. S. W. B. Prasetya, Frank Dignum, Mehdi Dastani, and Gabriele Keller. 2021. Using an Agent-Based Approach for Robust Automated Testing of Computer Games. In Proceedings of the 12th International Workshop on Automating TEST Case Design, Selection, and Evaluation (A-TEST ’21), August 23–24, 2021, Athens, Greece. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3472672.3473952 12. Dhia Elhaq Rzig, Nafees Iqbal, Isabella Attisano, Xue Qin, and Foyzul Hassan. 2023. Virtual Reality (VR) Automated Testing in the Wild: A Case Study on Unity-Based VR Applications. In Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2023). Association for Computing Machinery, New York, NY, USA, 1269–1281. https://doi.org/10.1145/3597926.3598134 |
| 指导教师意见 |
| 该同学所研究的VR APP的自动化异常检测工具着眼VR前沿领域，基于LLM、强化学习和寻路算法等对智能体进行设计与优化，同时通过实验开展分析研究工作，具有很好的研究价值。该同学前期已经积累了相关LLM、强化学习和程序检测的基础知识，并对相关国内外研究现状做了充足的调研工作，研究内容明确，研究方案与计划详实合理，同意开题。  指导教师：  2025年 1月 9 日 |
| 院（系）审查意见 |
| 同意开题。  学院领导（公章）：  2025年 月 日 |