

雷达虚拟操作交互方法设计与实现

马士友¹ 李毓辉¹ 卢亚飞¹ 常小龙¹ 倪翔¹

MA Shiyu LI Yuhui LU Yafei CHANG Xiaolong NI Xiang

摘要

针对雷达虚拟操作训练中存在的漫游控制与雷达操作抢占交互设备资源、雷达屏幕界面内容像素损失严重、人机交互误操作率高等问题,文章设计了一种模式分控、映射模式和局部漫游机制,以此适应不同的操作场景需求,达到交互设备资源优化利用、雷达屏幕界面内容像素无损、操作训练效率和沉浸感提升的目的,同时提出了一种共享热区的人机交互方案,以达到优化热区空间、降低操作难度和误操作率的目的。这两种技术方案设计思路简洁,技术路线简单易实现,示例基于 Unity3D 平台实现,但设计思路与方法也可以用于其他开发平台。

关键词

雷达;虚拟操作;人机交互;Unity3D

doi: 10.3969/j.issn.1672-9528.2026.06.020

0 引言

随着各类新型空中侦察与突击装备不断发展,作战环境与作战模式发生了深刻变化。为有效应对各类空中目标威胁,雷达作为核心对空探测预警手段,已成为提升防空预警与整体作战能力的关键支撑。与此同时,计算机技术、数字化技术、智能化技术在雷达领域的广泛应用,软件定义雷达(software-defined radar, SDR)成为雷达的未来发展方向之一^[1]。充分发挥雷达的作战能力,作为战争中最关键的因素,雷达操作使用人员必须具备高超的雷达操作技能。新型雷达技术先进,结构复杂,操作使用专业性更强,对训练的要求也相应提高。由于使用实装或全实物模拟训练器材进行雷达操作训练成本高,训练容量小,特别是在训练的初级阶段,训练的效费比较低,所以利用计算机以全虚拟形式进行雷达模拟操作训练,是目前费效比较高的一种训练方式^[2-3]。

1 雷达虚拟操作训练中存在的问题

雷达操作虚拟训练重点解决流程正确性、动作规范性、应用场景覆盖性问题,因此并不过度追求模拟训练系统物理结构与实装完全一致,但必须保证人机交互界面与真实界面保持一致。全虚拟模拟训练系统一般采用鼠标键盘控制,以第一人称视角漫游的方式,在计算机构建的三维仿真场景中进行模拟操作,新型雷达的交互设备一般包括雷达屏幕、键盘、鼠标(轨迹球)、各类开关按钮等,雷达操作人员通过鼠标点击等方式与之进行雷达操作,这些设备的虚拟仿真模型,通过模拟训练系统的显示设备显示出来。传统

的虚拟交互实现方式存在雷达屏幕界面内容像素损失严重、漫游控制与雷达操作抢占交互设备资源、人机交互误操作率高等问题^[4]。

1.1 漫游控制与雷达操作抢占交互设备资源

新型雷达的键盘与普通计算机键盘布局基本相同,模拟训练系统可以使用常规键盘、鼠标来直接进行雷达操作训练,但在进行漫游控制时,模拟训练系统一般采用第一人称射击游戏的操控方式,使用键盘控制虚拟人移动,使用鼠标控制虚拟人视角,漫游控制与雷达操作都需要使用鼠标和键盘,造成了相互抢占交互设备资源的情况。

1.2 雷达屏幕界面内容像素损失严重

新型雷达的屏幕分辨率都比较高,界面显示的内容非常丰富,除目标回波等信息外,还有很多软件菜单、按钮,雷达操作人员使用模拟训练系统在三维场景中漫游时,雷达屏幕在训练系统显示器上形成了“屏中屏”的情况,雷达屏幕界面内容像素损失严重,观察角度也难以调整到垂直角度,进一步导致产生透视畸变,其中的内容难以辨识,基本上无法进行软件菜单和按钮等交互操作。

1.3 人机交互误操作率高

雷达操作还包括很多机柜按键开关的操作,这些按键开关本身体积比较小,在交互操作时,如果使用鼠标点击进行交互操作,一是精准点击的难度比较大,容易误操作,二是难以判断鼠标是否在按键开关热区内,增加了交互的难度。这些问题使得雷达操作人员需分出部分精力关注交互操作动作,对雷达操作训练本身产生负面影响。

1. 陆军兵种大学 河南郑州 450052

2 雷达虚拟操作的人机交互设计

在院校或训练基地的场景下，雷达操作训练的主要特点是人多装备少，尽量用软件代替硬件开展训练是提高训练费效比的重要手段之一。通过优化雷达虚拟操作交互方法，可以在不增加硬件的前提下加强雷达模拟训练的真实感。

2.1 模式分控

针对雷达虚拟训练中漫游控制和雷达操作抢占交互设备资源的问题，可以采用模式分控的设计，所谓模式分控，是指设计两种基本操控模式——漫游模式和操作模式，两种模式可以互相切换，分别应对漫游控制和雷达操作的不同需求。这是利用雷达操作训练时漫游控制与雷达操作不同时发生的特点，即进行雷达操作时，操作人员基本保持静止状态，视角不变，在当前雷达操作步骤完成后，利用漫游模式进入下一个操作位置；在操作模式时，鼠标键盘从漫游模式退出，不再控制虚拟人的位置与视角，可以直接进行雷达操作。漫游与操作模式如图 1 所示。

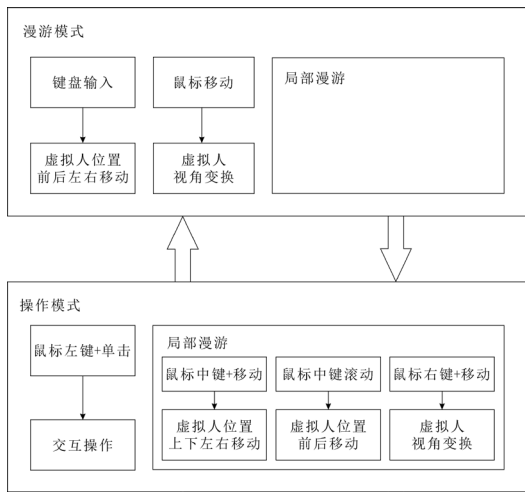


图 1 漫游与操作模式示意图

2.2 映射模式

在进行雷达屏幕界面操作时，会出现前文所说“屏中屏”的问题，为了消除雷达屏幕界面内容像素损失问题，在操作模式中设计一种映射模式，即将雷达屏幕的内容直接映射显示到模拟训练系统的显示设备上，以实现像素级点对点的显示效果，便于观察和操作，如图 2 所示。通过自定义快捷键，切入到设备映射状态，同时，将雷达的键盘鼠标（轨迹球）与模拟训练系统的键盘鼠标进行映射，此时，模拟训练系统的鼠标可以直接控制雷达显示器上的光标移动和点击，键盘则可以直接在雷达软件上进行输入或使用快捷键进行软件操作。在进行设备映射时，雷达屏幕已经全部映射到模拟训练系统的显示设备上，此时不再有局部漫游的需求，需要屏蔽局部漫游模式，以避免操作产生冲突。

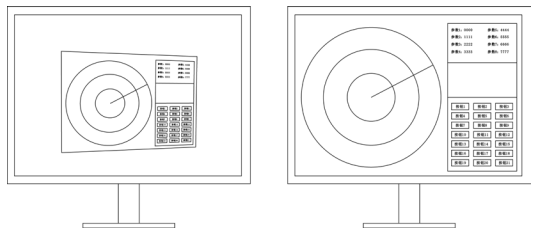


图 2 “屏中屏”与点对点映射对比示意图

从使用层面看，映射模式属于操作模式，但是从软件设计层面看，映射模式下应屏蔽一切漫游操作和除雷达键盘鼠标（轨迹球）输入以外的其他交互操作，所以漫游模式、操作模式、映射模式三者之间在软件设计中为互斥关系，且可以自由切换，如图 3 所示。

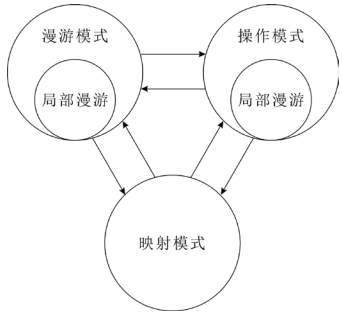


图 3 三种模式关系示意图

2.3 局部漫游

从雷达操作训练实际来看，操作模式下也有局部位置移动和视角变换的需求。此时，如果频繁切换状态，操作复杂、效率较低，影响训练效果。经过实践探索发现，如果设计“局部漫游”功能，即利用在漫游模式和操作模式两种模式下都空闲的鼠标按键，进行局部位置移动和视角变换控制，通过软件参数设置，可以控制位置和视角的移动速度，提高鼠标的控制精度，降低操作难度，提高操作效率。

2.4 交互增强

为解决热区过小导致的交互错误率高的问题，从两个方面进行了交互增强设计：一是共享热区设计。通常情况下，使用鼠标在操作某一操作对象时，需要鼠标悬停在该对象上，方能进行交互操作，即交互热区为操作对象本身。当操作对象比较小时，鼠标准确悬停在热区上的难度增加，可以在不影响相邻操作对象的前提下，适当增大交互热区，以降低操作时鼠标定位的难度。但在雷达模拟训练中，机柜按键开关较多，部分按键开关设计紧凑，为避免相互重叠影响检测，热区间不可避免地要留有空隙，热区空间不能最大化利用。所谓共享热区，即人工将相邻的按键开关划分为一组，共享同一个热区，如图 4 所示。按钮 1～6 共享热区 A，按钮 7～9 共享热区 B。在鼠标进入热区后，距离鼠标最近的对象响应鼠标操作；二是人工回路设计。在鼠标在进入热区悬停时，

增加视觉或听觉反馈,将操作人员纳入判别回路中来判断鼠标是否已经进入该操作对象的热区中,从而增加操作的准确性,降低误操作率。

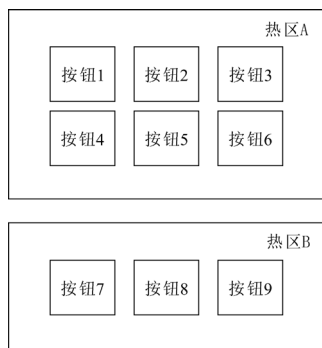


图4 共享热区示意图

3 基于 Unity3D 的雷达操作交互实现方法

Unity3D 是一款广泛使用的游戏开发引擎,具有可视化编辑工具和强大的 C# 脚本支持,支持多平台开发和跨平台发布,广泛应用在游戏、影视、工业仿真等领域。

3.1 定义模式函数与控制变量

在 Unity3D 脚本中定义 4 个函数:漫游函数 ManYou(),交互函数 CaoZuo()、局部漫游函数 JuBu() 和映射函数 YingShe(),全部在系统函数 Update 内进行调用。

定义一个 byte 变量 moShi,默认值为 0,执行 ManYou() 和 JuBu() 函数,进入漫游模式,当 moShi 值为 1 时,执行 CaoZuo() 和 JuBu() 函数,进入操作模式,当 moShi 值大于等于 2 时,执行 YingShe() 函数,进入映射模式,函数逻辑关系如图 5 所示。

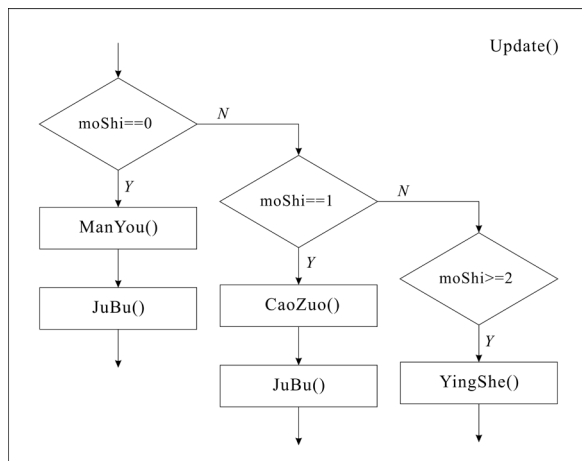


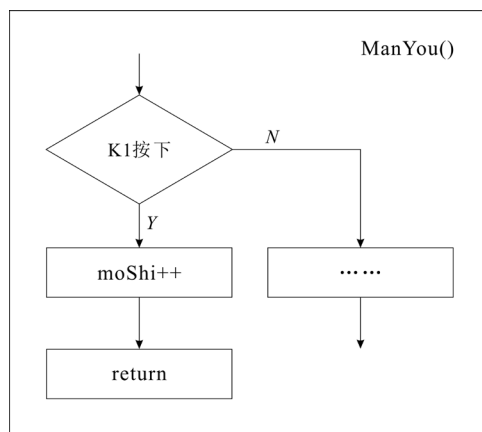
图5 模式函数及执行逻辑关系

3.2 模式切换控制

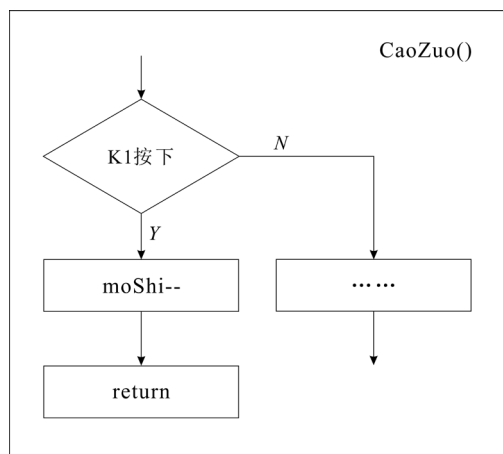
在 Unity3D 中切换模式可以通过自定义快捷键实现,根据雷达操作内容,选择两个空闲快捷键,假设为 K1 和 K2,作为自定义模式切换快捷键。

3.2.1 漫游模式与操作模式的切换控制

在漫游模式下,moShi 值为 0,执行漫游函数 ManYou(),在该函数中对 K1 键进行监听,当按下 K1 快捷键时,moShi 值加 1,由 0 变为 1,执行交互函数 CaoZuo(),切换至操作模式。在交互函数 CaoZuo() 中,对 K1 键进行监听,当按下 K1 快捷键时,moShi 值减 1,由 1 变为 0,执行漫游函数 ManYou(),切换回漫游模式,以实现漫游和操作模式的相互切换。控制程序如图 6 所示。



(a) 漫游模式向操作模式切换

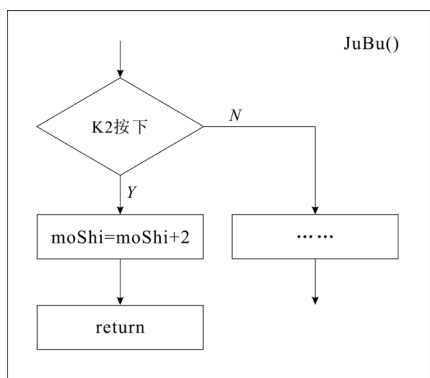


(b) 操作模式向漫游模式切换

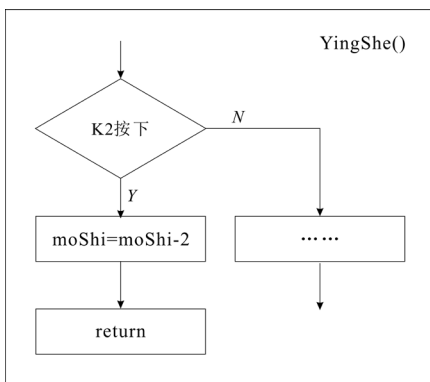
图6 漫游模式与操作模式切换控制程序示意图

3.2.2 漫游模式 / 操作模式与映射模式的切换控制

无论是漫游模式还是操作模式,都可以一键切换至映射模式。漫游和操作模式都执行局部漫游函数 JuBu(),在 JuBu() 中对 K2 快捷键进行监听,当按下 K2 快捷键时,moShi 值加 2,此时 moShi 值由 0 或 1 变为 2 或 3,执行映射函数 YingShe(),切换至映射模式。在映射函数 YingShe() 中,对 K2 键进行监听,当按下 K2 快捷键时,moShi 值减 2,此时 moShi 值由 2 或 3 变为 0 或 1,执行漫游函数 ManYou() 或交互函数 CaoZuo(),实现返回至前一模式,控制程序如图 7 所示。



(a) 漫游模式 / 操作模式向映射模式切换



(b) 映射模式向漫游模式 / 操作模式切换

图 7 漫游模式 / 操作模式与映射模式切换控制程序示意图

3.3 映射模式实现

3.3.1 Unity3D 相关组件创建

在 Unity3D 中，创建实现雷达屏幕映射效果所需相关组件，组件类型和功能如表 1 所示。

表 1 雷达屏幕映射相关 Unity3D 组件

组件名称	类型	功能
MainCamera	Camera	主摄像机，用于渲染主视角画面
UICamera	Camera	雷达屏幕摄像机，用于渲染雷达屏幕画面
PingMuCV	Canvas	画布，用于绘制雷达屏幕 UI 元素
PingMu	Cube	长方体组件，用于模拟雷达屏幕实体
PingMuRender	Render Texture	纹理，用于映射 UICamera 画面

3.3.2 映射模式实现方法

为了在屏幕模拟组件上能够实时显示雷达屏幕的内容，需要进行以下设置：

画布 PingMuCV 的 Canvas 组件中 Render Mode 项设置为 Screen Space-Camera 模式，Render Camera 项设置为 UI-Camera。

摄像机 UICamera 的 Camera 组件中 Target Texture 项设置为 PingMuRender。

将渲染纹理 PingMuRender 赋予雷达屏幕 PingMu 的材质项。

通过以上设置，建立起组件间的逻辑联系，从而完成屏幕内容从画布到屏幕模拟组件的传递，各组件之间的逻辑关系如图 8 所示。

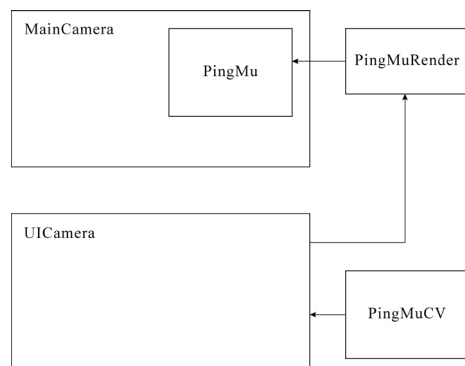
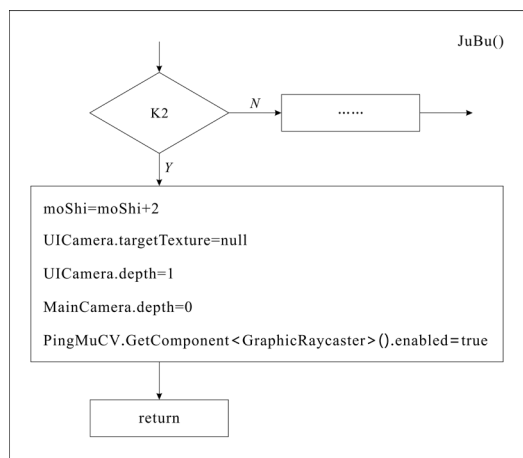


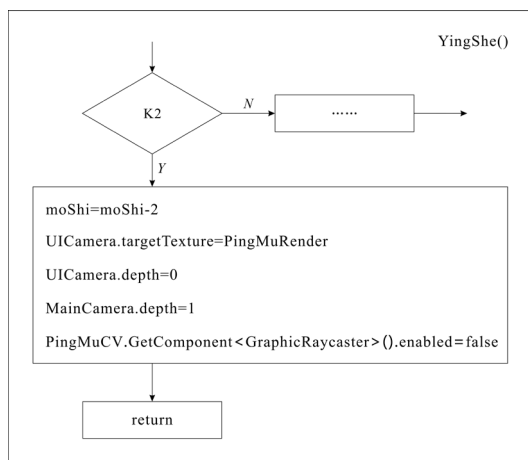
图 8 映射模式组件逻辑关系图

3.3.3 摄像机切换方法

要实现映射模式的切换，需要控制主摄像机 MainCamera 和雷达屏幕摄像机 UICamera 的内容在模拟训练系统显示设备上的显示与隐藏。在 Unity3D 中，摄像机的渲染顺序由 Camera 组件中的 Depth 值决定，值越大渲染优先级越高，可以通过更改两个摄像机的 Depth 值来确定显示哪一个摄像机内容。需要注意的是，UICamera 因为需要向渲染纹理 PingMuRender 输出显示内容，在此设置下，不能同时向显示设备输出画面，所以当映射模式时，要更改雷达屏幕摄像机 UICamera 的 Camera 组件中的 Target Texture 为空，以保证画面的正常输出。另外当非映射模式时，屏幕摄像机 UICamera 的内容虽然被主摄像机遮盖，但画布 PingMuCV 中的 UI 仍然会响应鼠标动作，为了防止在非映射模式下 UI 响应鼠标动作产生误操作，在切换至非映射模式时，需要禁用画布 PingMuCV 的交互组件——Graphic Raycaster（图形射线投射器）^[5]。摄像机切换程序示意图如图 9 所示。



(a) 主摄像机切换为雷达屏幕摄像机



(b) 雷达屏幕摄像机切换为主摄像机

图9 摄像机切换程序示意图

3.4 交互增强

3.4.1 创建碰撞检测组件

在 Unity3D 中，热区的实现可以使用碰撞检测组件 Collider，Collider 系统内置了一些基本类型，如球体、长方体等，可以满足绝大多数的场景。首先是按照需要交互的按钮开关等操作对象的分布，按照距离相近、功能相关的原则进行分组，每组创建一个空物体并添加碰撞检测组件 Collider，调整 Collider 的大小，使之覆盖该组内所有操作对象，并根据具体适当增大其大小，且以不与相邻 Collider 重叠为限。

3.4.2 创建碰撞检测脚本

创建碰撞检测脚本，挂载在用于分组的空物体上，脚本流程示意图如图 10 所示。

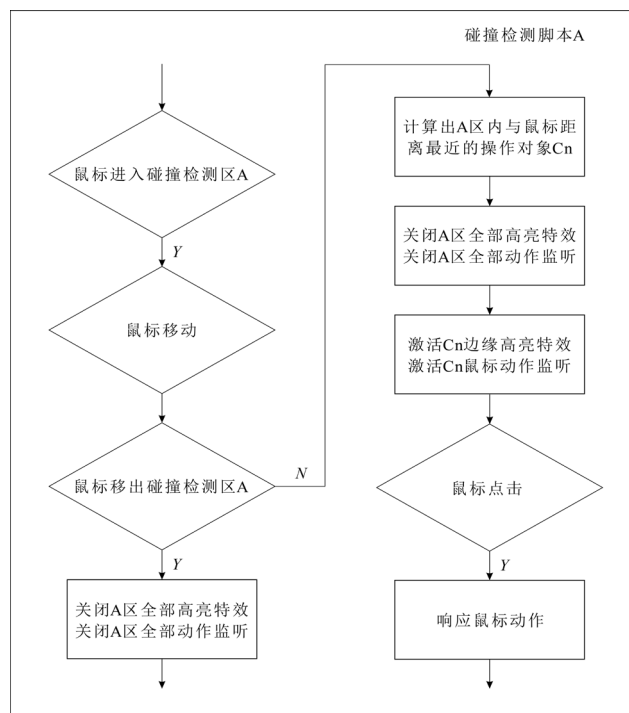


图10 碰撞检测流程示意图

监听鼠标是否进入碰撞检测区 A，如果鼠标进入且有移动动作，则计算鼠标与热区 A 内所有操作对象中心的距离，找出距离鼠标最近的操作对象 Cn 进行高亮显示，并激活 Cn 的鼠标动作监听，当鼠标有点击动作时，响应该对象对应的操作动作。如果鼠标移出热区 A，则关闭热区 A 内所有操作对象的高亮显示和鼠标动作监听。

4 结语

本文针对院校和训练基地应用场景，对雷达虚拟操作交互方法进行了优化设计，以计算机的一般操作为基础，结合传统第一人称游戏操作习惯和主流专业三维建模软件中的场景控制方式，设计了两种人机交互技术方案，能有效降低操作者的学习和适应成本，降低操作难度，提升操作效率和准确率，加强训练沉浸感，提高训练效率。本文提出的技术方案设计思路简洁，技术路线简单易实现，文中示例基于 Unity3D 平台实现，但设计思路与方法也可以用于其他开发平台。

参考文献：

- [1] 邱磊, 杨君, 朱卫纲. 某型相控阵雷达模拟训练系统设计与实现 [J]. 科技与创新, 2024(20):53-56.
- [2] 李华君, 周丽, 胡天皓. 基于交互认知的雷达显控设计方法 [J]. 雷达与对抗, 2023,43(4):49-53.
- [3] 武忠国, 赵宏宇, 李坤, 等. 基于虚实结合的雷达操作训练方法研究 [J]. 指挥控制与仿真, 2024,46(3):130-136.
- [4] 潘奇, 杨东华, 蔡委哲. 自然人机交互技术在雷达显控终端中的应用 [J]. 现代雷达, 2024,46(8):78-83.
- [5] 宣雨松. Unity3D 游戏开发 (第三版) [M]. 北京: 人民邮电出版社, 2023.

【作者简介】

马士友 (1981—), 男, 黑龙江汤原人, 硕士, 讲师, 研究方向: 虚拟仿真、模拟训练。

(收稿日期: 2025-11-18 修回日期: 2026-06-01)