《计算机图形学》实验报告

1. 实验内容
2. 实验目的

熟悉如何在VSCODE中配置freeglut库，并能够在此环境中进行OpenGL编程。

通过具体的编程任务，实践并加深对OpenGL和freeglut库功能的理解，包括基本绘图、变换、动画等。

通过编写机器人在不同场景中（如走到床边、坐到椅子上、关闭电视机等）的动画代码，理解动画逻辑的设计和实现。

在完成基本任务的基础上，实现额外的动画效果（如视角变换、机器人走出户外等）。

1. 常见解决方法

基于关键帧的动画（Keyframe Animation）。通过设定关键时间点的关键帧，定义每个时间点上动画对象的位置和状态，利用插值计算中间帧的状态。

骨骼动画（Skeletal Animation）。使用一组骨骼来控制模型的变形，每个骨骼通过关节连接，动画通过旋转和移动骨骼来实现。

基于物理的动画（Physics-Based Animation）。利用物理模拟（如力学、碰撞检测等）计算动画对象的运动轨迹，使其运动更加真实。

路径动画（Path Animation）。通过预定义的路径来控制动画对象的运动，动画对象沿路径移动。

1. 优缺点比较

基于关键帧的动画（Keyframe Animation）的优点是，简单直观，容易实现和控制，适合需要精确控制的动画；缺点是，复杂动画定义关键帧较繁琐；插值计算可能导致非自然的运动。

骨骼动画（Skeletal Animation）的优点是，适合复杂和多关节的模型动画，存储和计算效率高；缺点是，实现复杂，需要较高的数学和编程基础，设计骨骼系统需要经验。

基于物理的动画（Physics-Based Animation）的优点是，动画效果真实自然，适合模拟自然现象；缺点是，计算量大，实时性差，需要复杂的物理建模和求解。

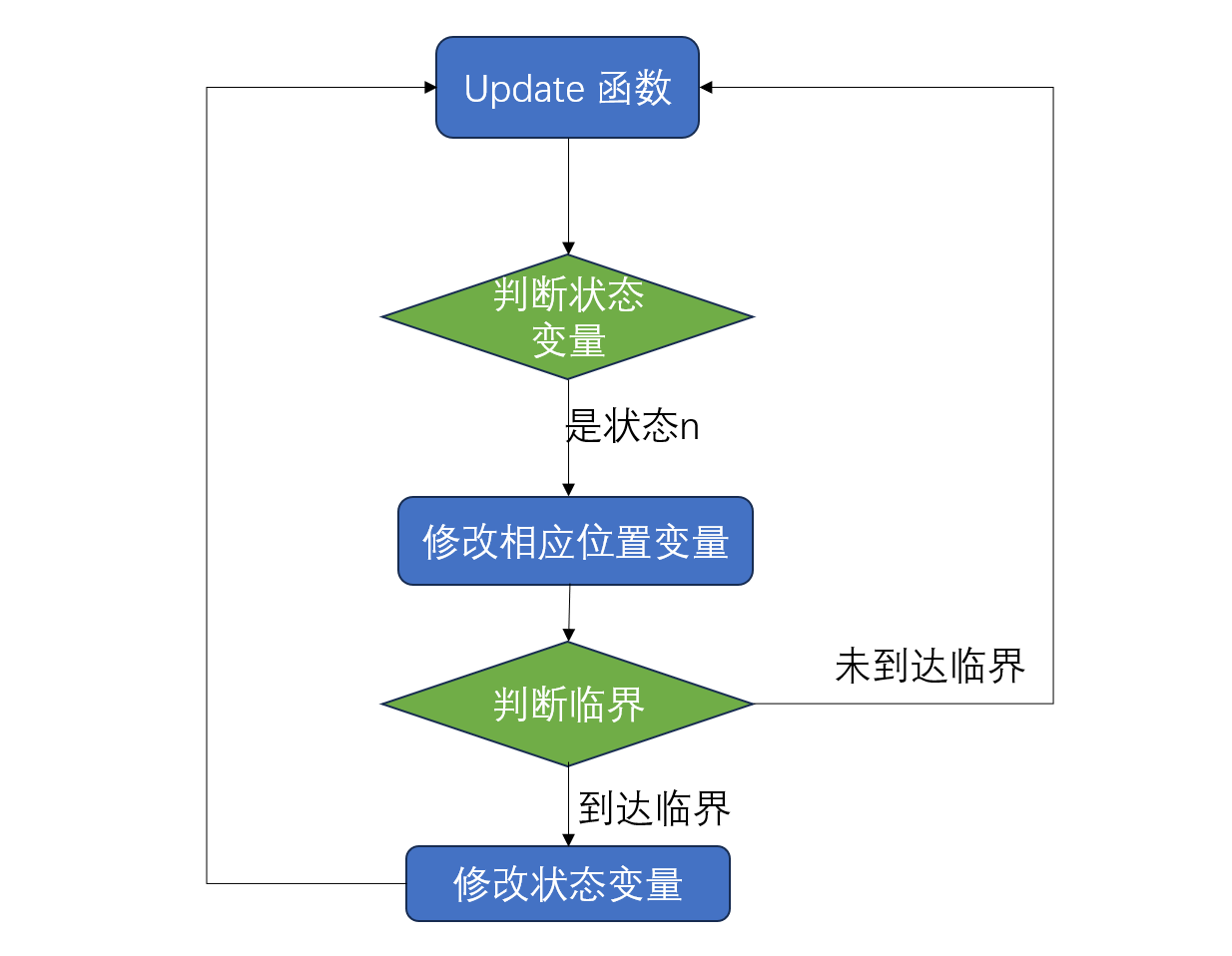
路径动画（Path Animation）的优点是，实现简单，适合规则运动轨迹，控制方便，路径设计清晰；缺点是，灵活性差，不适合复杂运动，路径设计需要细致考虑。

1. 实现方法

1. 方法原理

调用OpenGL的绘图函数和变换函数，可以绘制人物的各个部分（如头、身体、四肢等）并对其进行平移、旋转等变换。通过调整全局变量，可以实现人物的不同姿态和动作。update函数用于更新人物的状态，使其在3D空间中进行移动、旋转等动作，实现简单的动画效果。

2. 程序设计思路



初始化和全局变量：全局变量用于跟踪人物的位置、姿态和状态。位置变量：x\_position、y\_position、z\_position，用于记录人物在3D空间中的位置。角度变量：arm\_angle、leg\_angle、rotation\_angle、sitting\_angle，用于控制人物的手臂、腿的摆动，整体旋转，以及坐下的角度。状态变量：walking1、walking2、sitting、arm\_up、sleeping，用于记录人物当前的动作状态，如行走、坐下、抬手、睡觉等。

状态更新函数（update函数）是动画系统的核心，负责根据当前状态和时间步长更新人物的位置、姿态等信息。该函数的主要作用包括：更新位置和角度，根据不同的状态（如walking1、walking2等），逐步更新人

物的位置和各个关节的角度。状态切换，在特定条件下（如达到一定的位置或角度），切换人物的状态，例如从行走切换到坐下、抬手等。具体而言，当人物处于行走状态时，update函数会逐步增加x\_position和z\_position，并相应地摆动手臂和腿。当人物达到目标位置后，切换到睡觉状态或其他状态。类似地，当人物处于坐下状态时，update函数会逐步增加sitting\_angle，直到达到预期角度，然后切换到抬手状态（关闭电视姿势）。在关闭电视后，走向传送门位置，当接近传送门时，传送门显形。到达室外，完成飞行和降落动作。

3. 程序重难点解释

场景切换是动画系统中的一个重要功能，涉及从一个场景（如室内）切换到另一个场景（如室外）。需要处理与新场景相关的状态和变量。在实现场景切换时，需要考虑，场景初始化。每个新场景需要初始化特定的对象和变量，室内场景包含家具、墙壁等对象，而室外场景可能包含草地、树木、泳池、花丛等。在实际实现中，当人物从室内场景走到传送门前并触发切换条件时，update函数会调用相关的初始化函数加载室外场景，并更新全局变量以适应新场景。

人物运动包括行走、转身等动作。为了实现这些效果，update函数需要不断更新人物的位置和姿态。例如，在行走状态下，系统通过逐步增加x\_position和z\_position来模拟人物前进。同时，通过摆动手臂和腿的角度，模拟出行走的动态效果。当达到目标位置时，系统会切换到下一个状态，如转身或坐下。

实现人物睡觉状态需要控制人物的姿态和位置。在睡觉状态下，系统需要调整人物的姿态，使其看起来像躺下的状态。在update函数中，逐步

增加sleep\_time，模拟睡觉效果。

坐下动作需要控制身体各个部分的角度，尤其是腿部和身体的弯曲角度。在update函数中，逐步增加sitting\_time，模拟出坐下的过程。当达到设定时间时，系统会切换到抬手（关闭电视机）。

1. 结果分析
2. 实验环境

设备名称：LAPTOP-418TSMAV

处理器：11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz

机带RAM ：16.0 GB (15.8 GB 可用)

系统类型：64 位操作系统, 基于 x64 的处理器

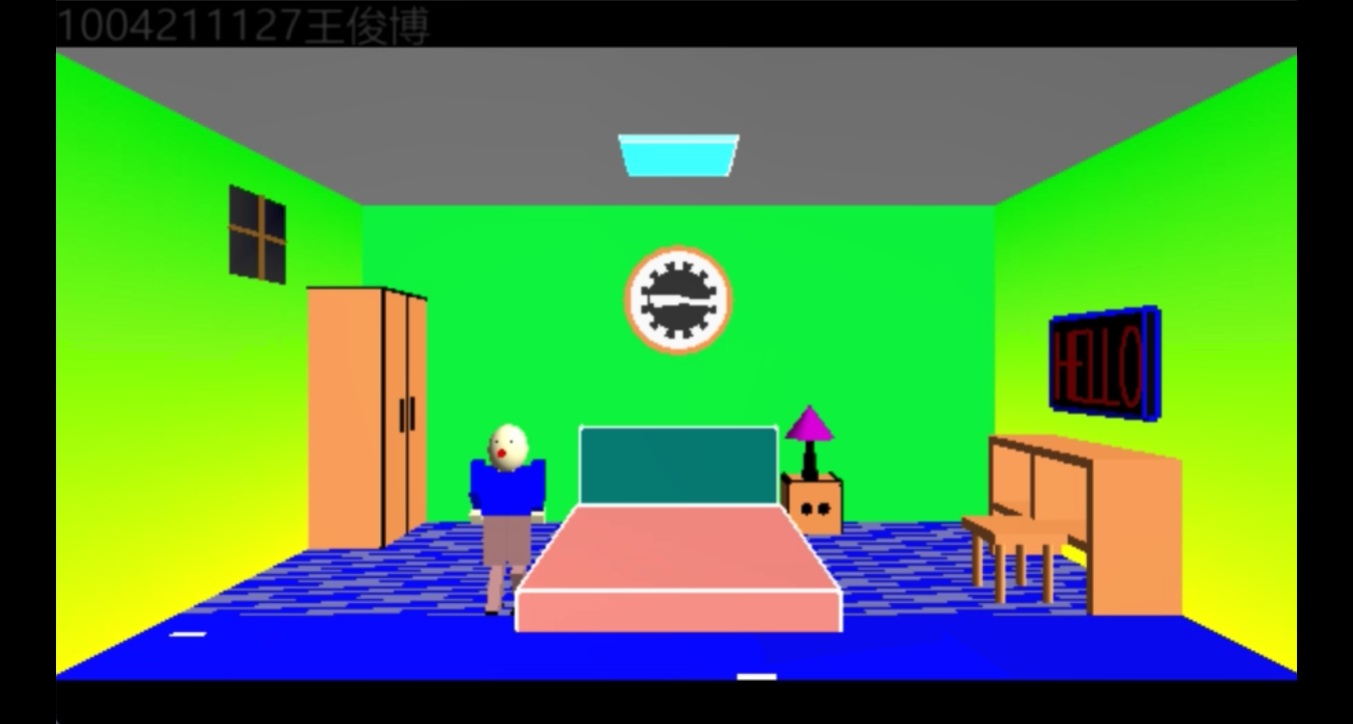
集成开发环境（IDE）：Vscode

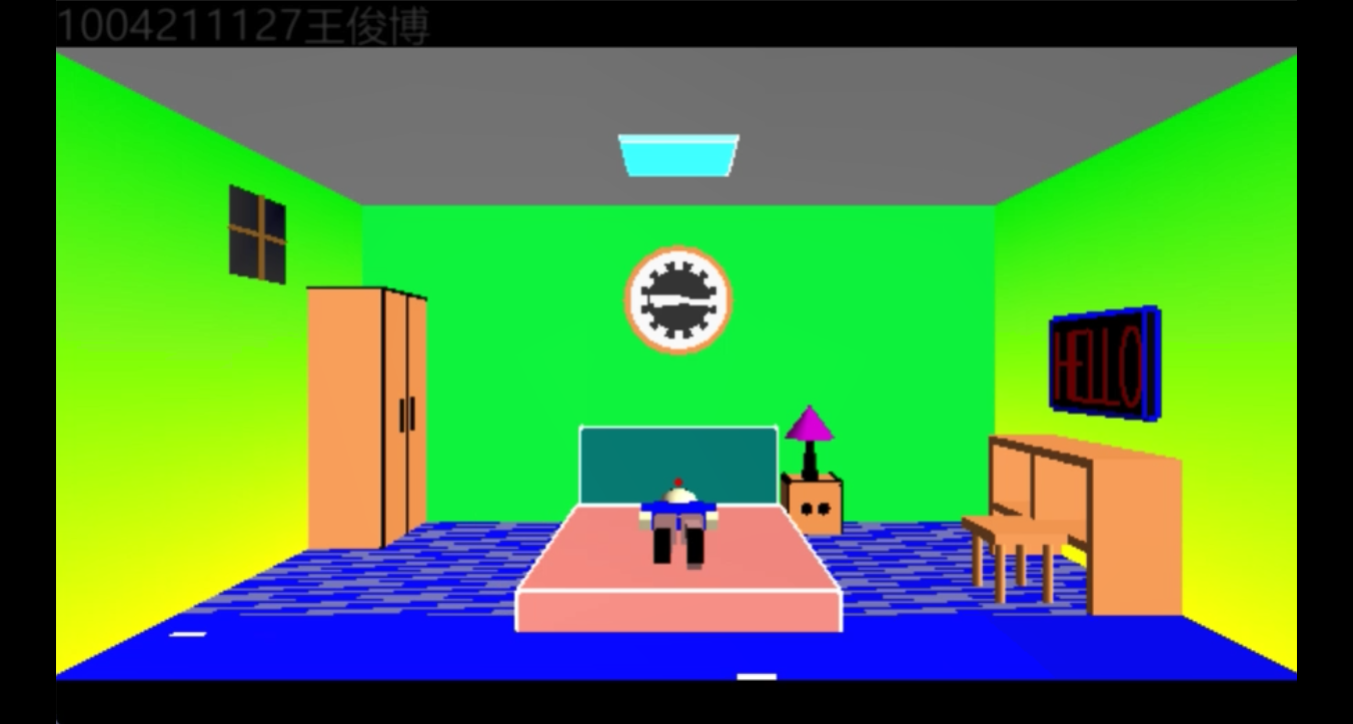
库：freeglut

编程语言：C++

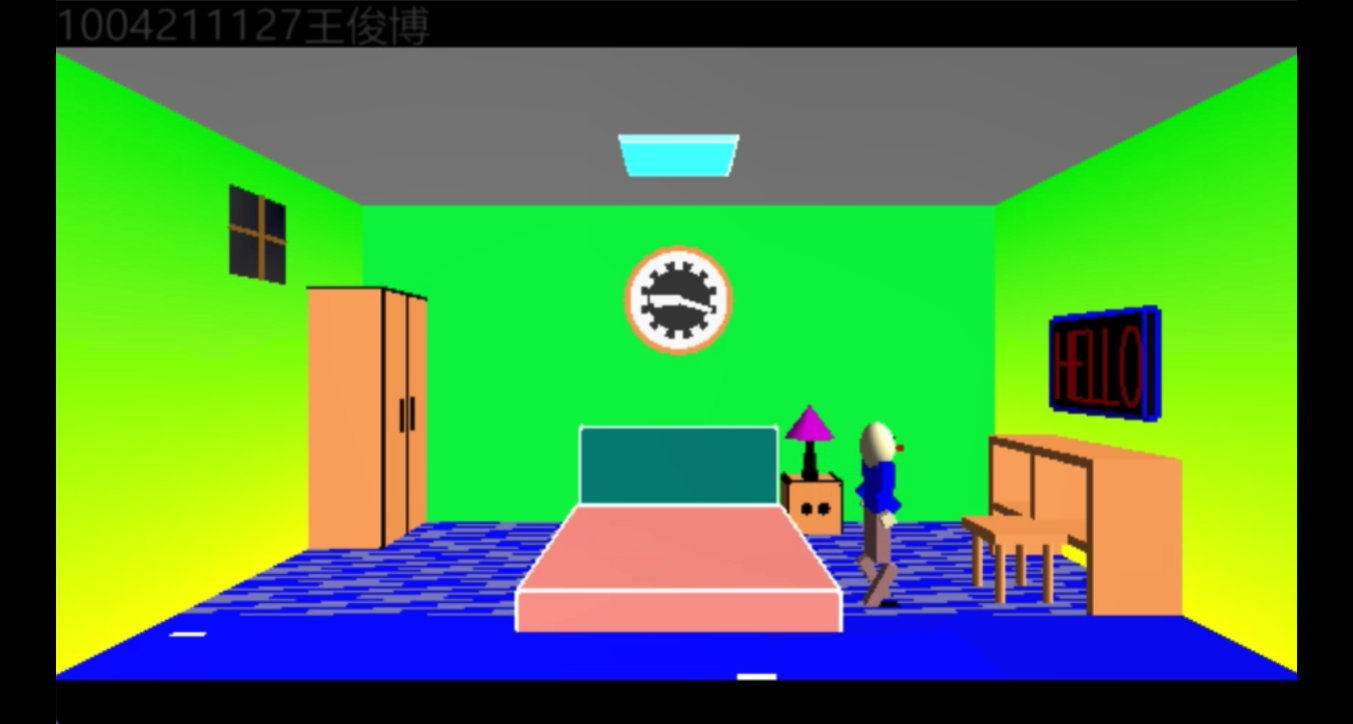
1. 实验结果

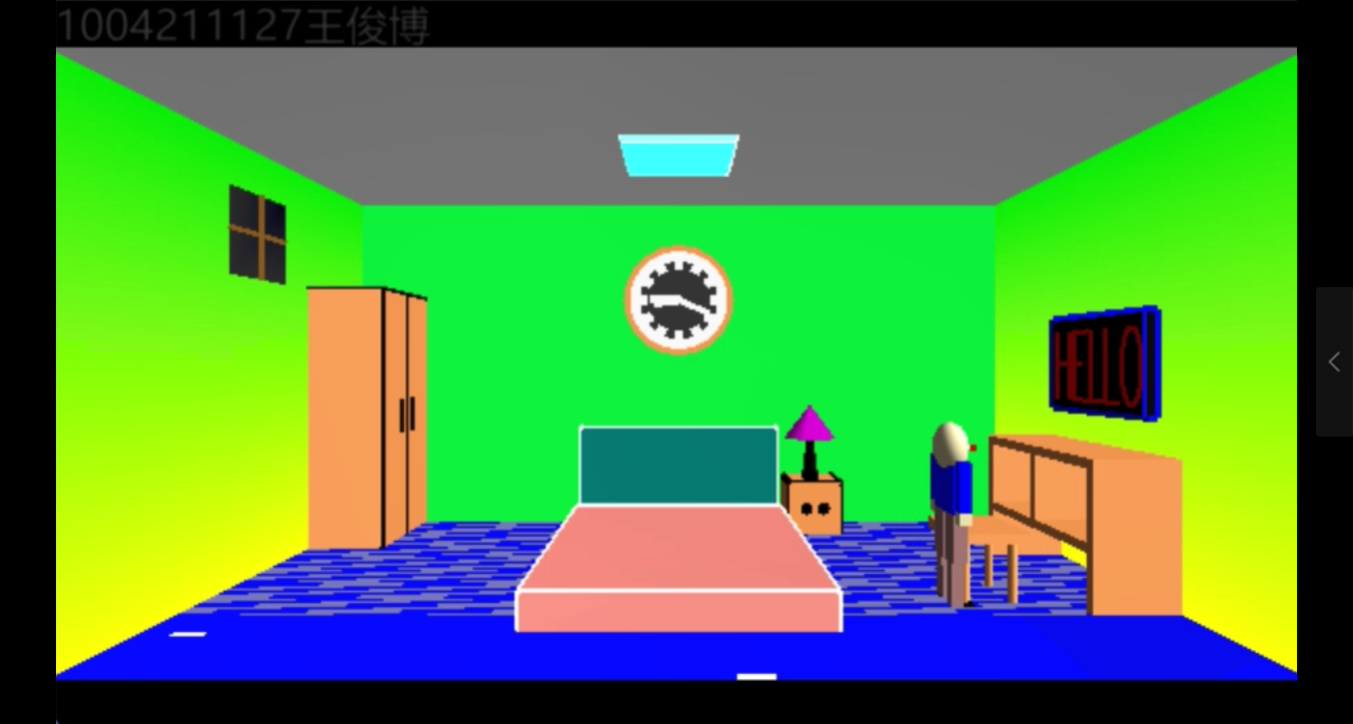
机器人走到床边躺下

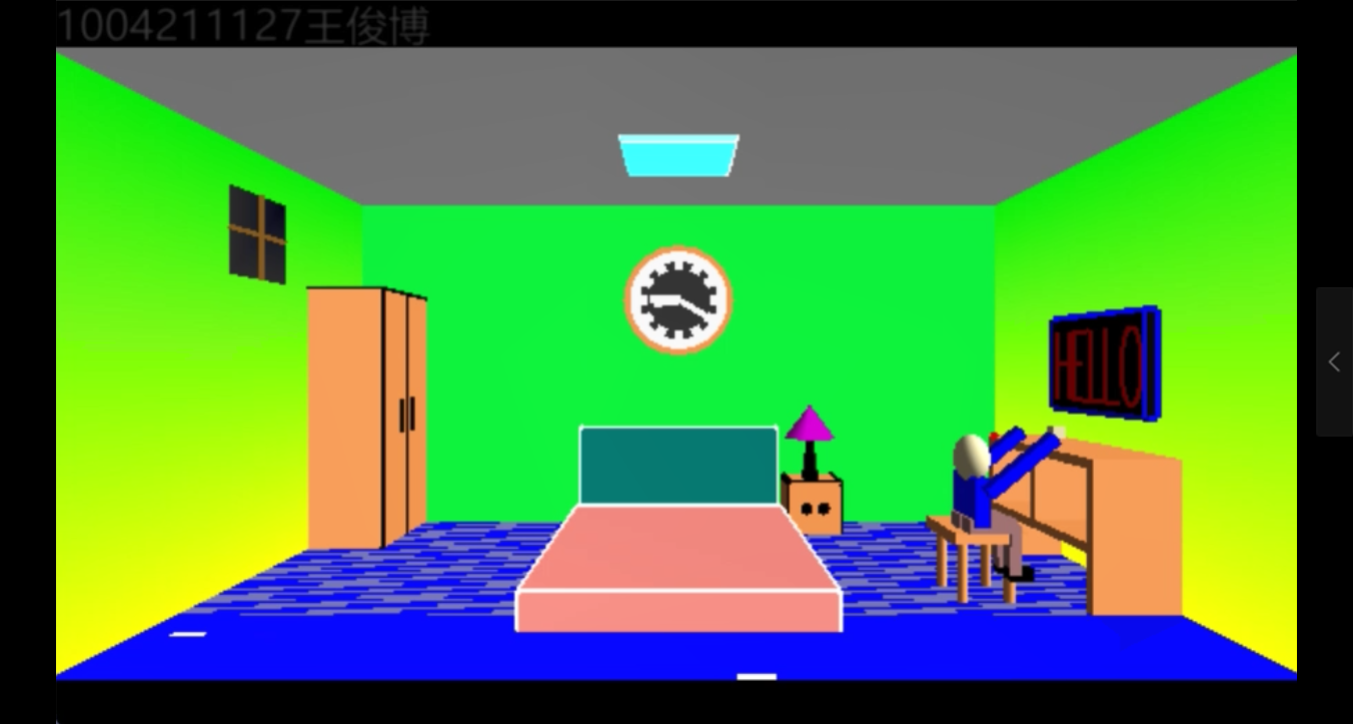




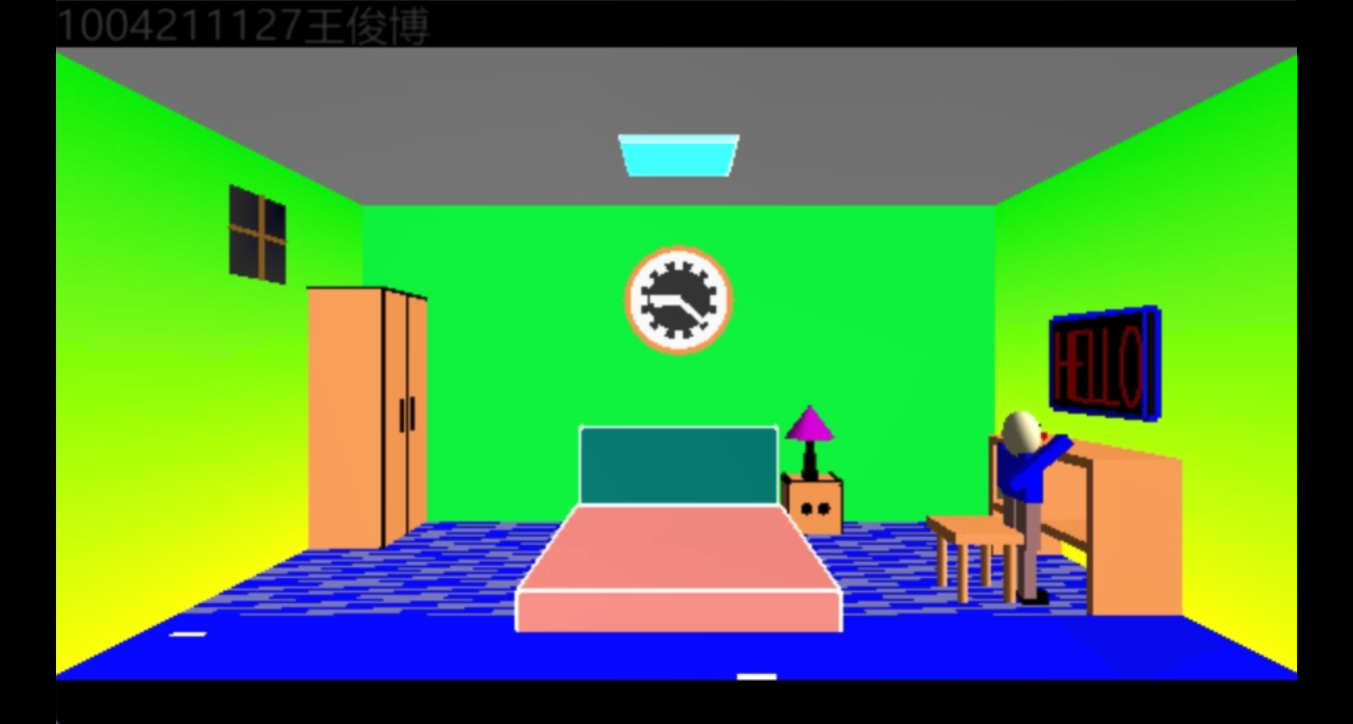
机器人走到椅子边拉出椅子，坐在椅子上，双臂放在桌子上

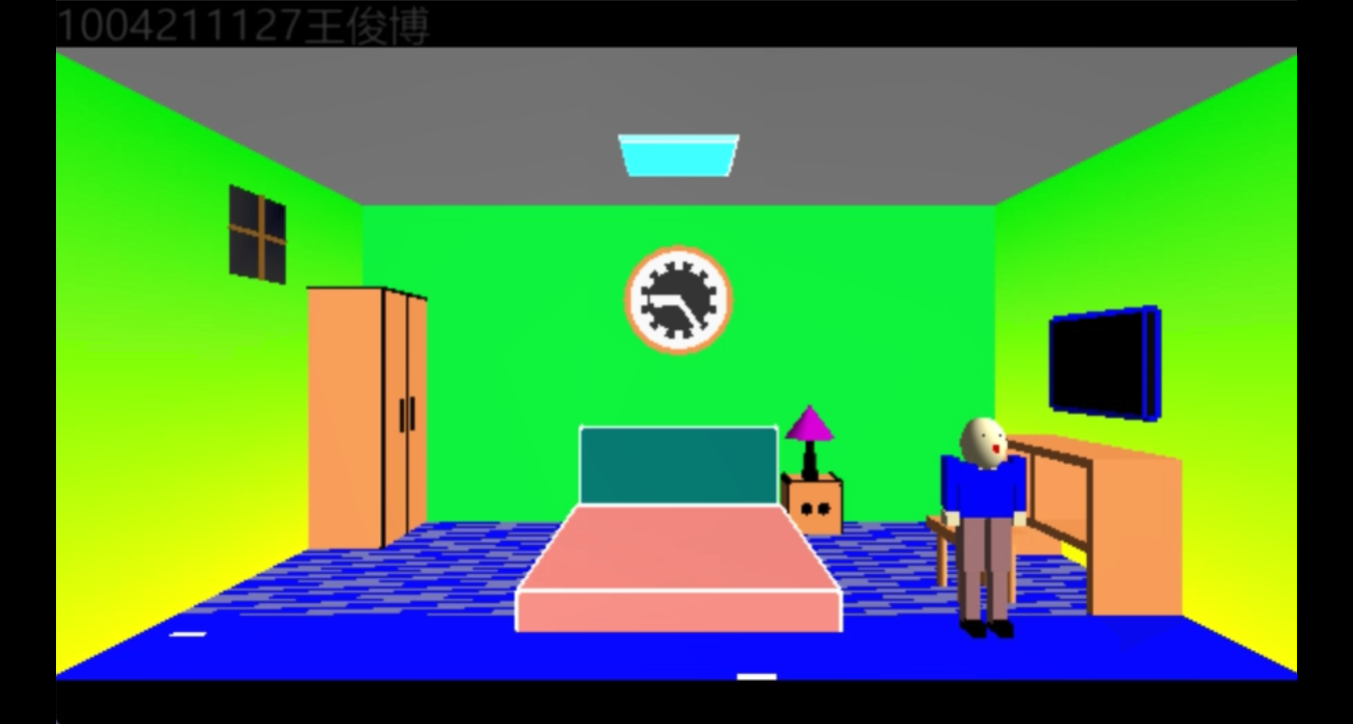




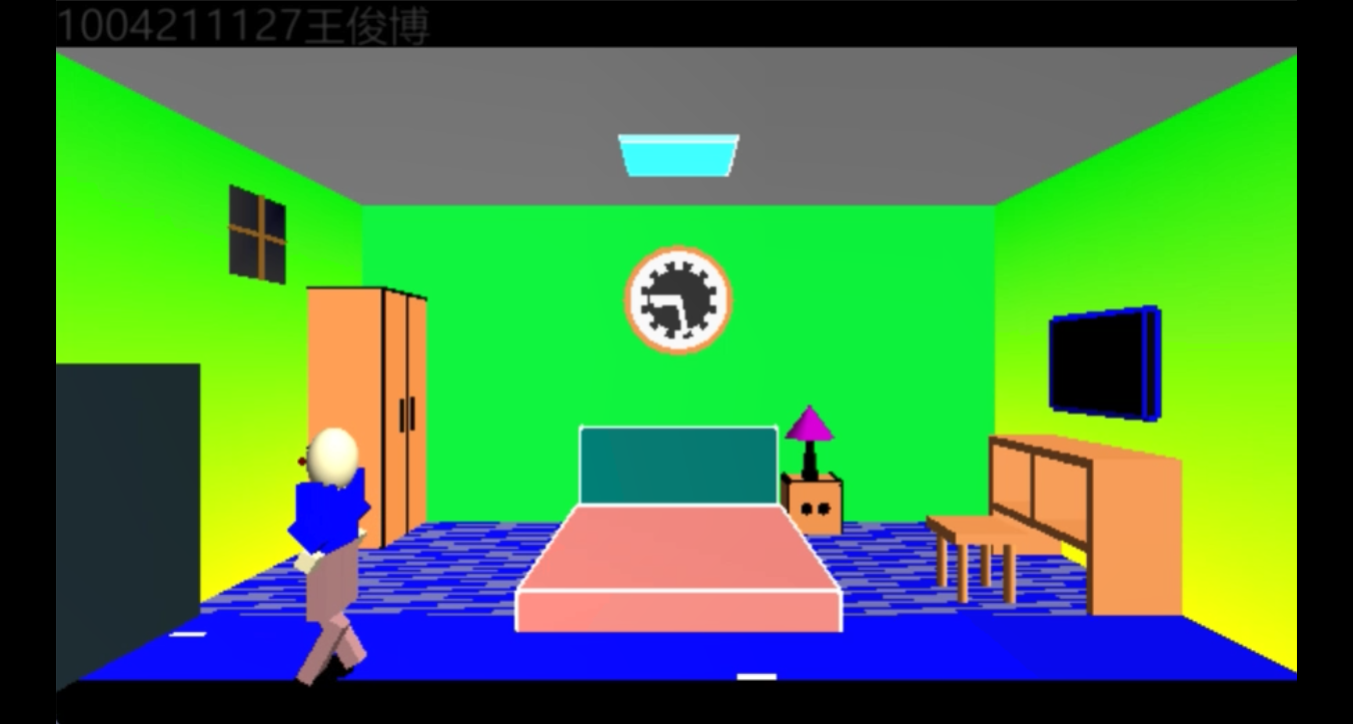


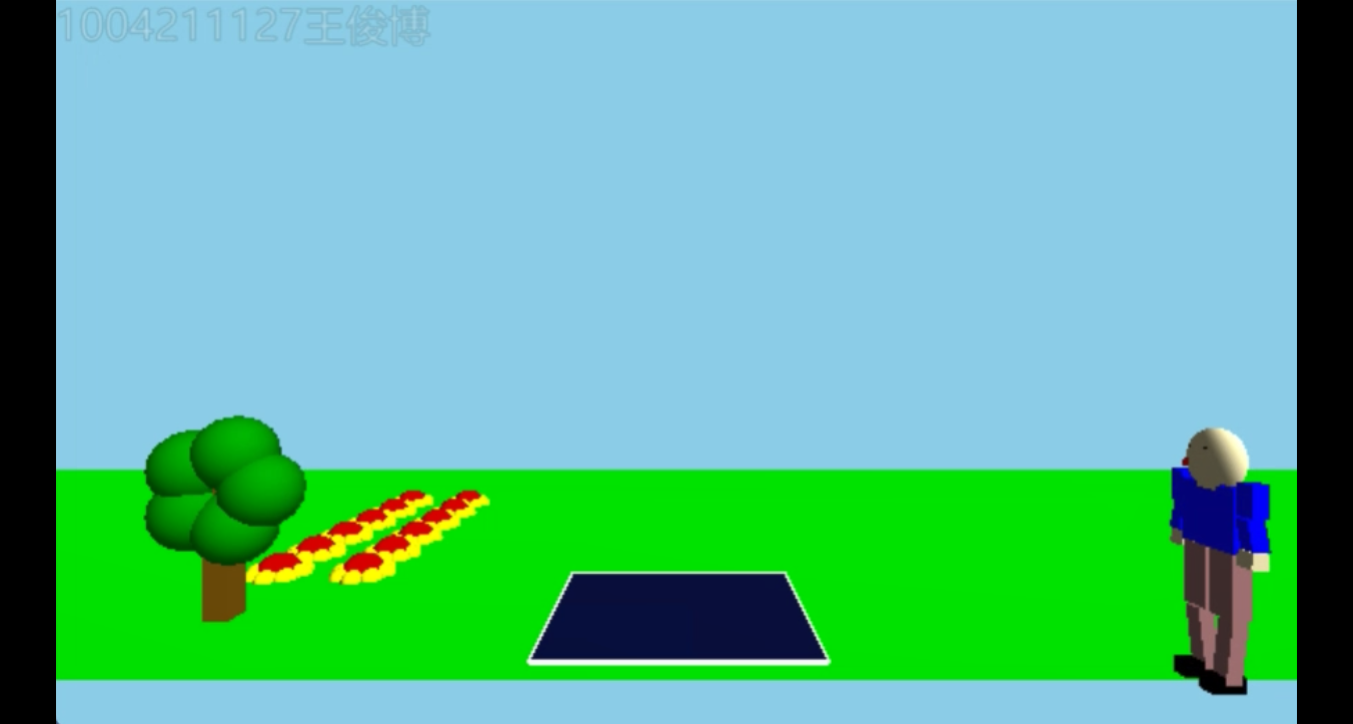
机器人关闭电视机的动作和关闭电视机后的效果图



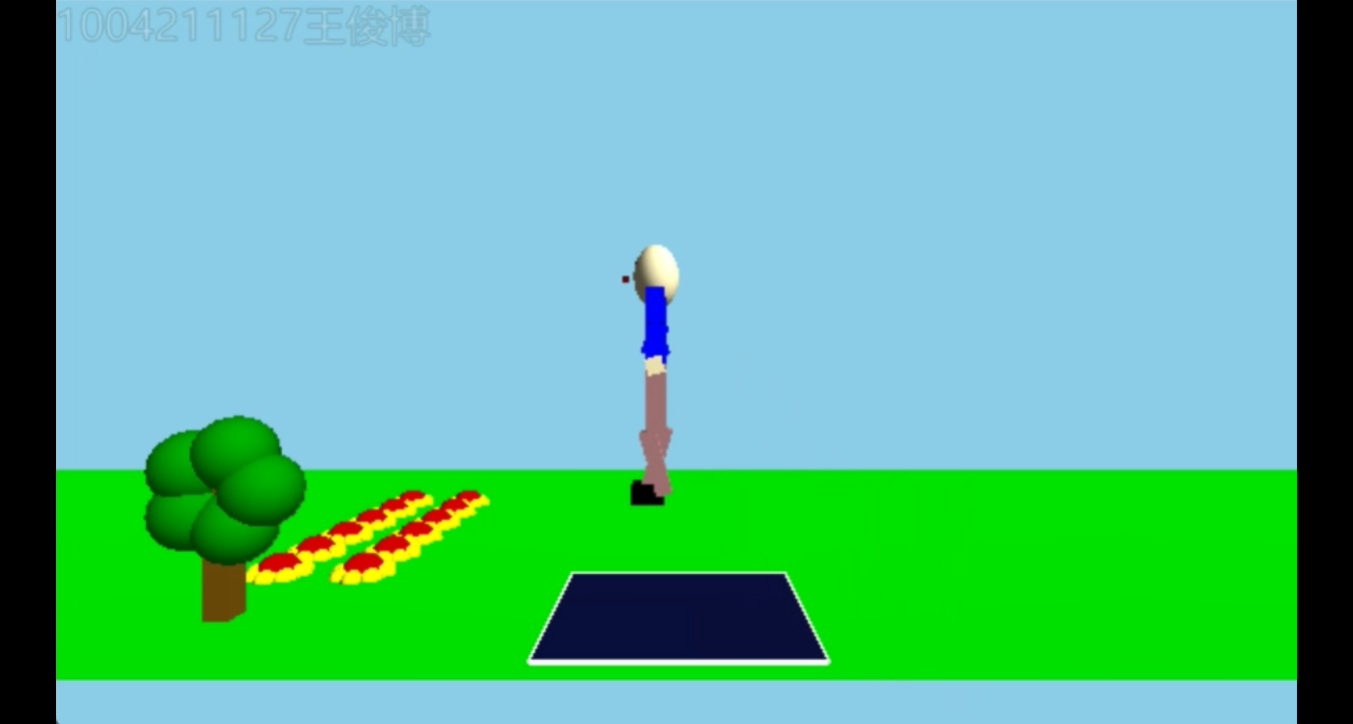


机器人走出户外看风景

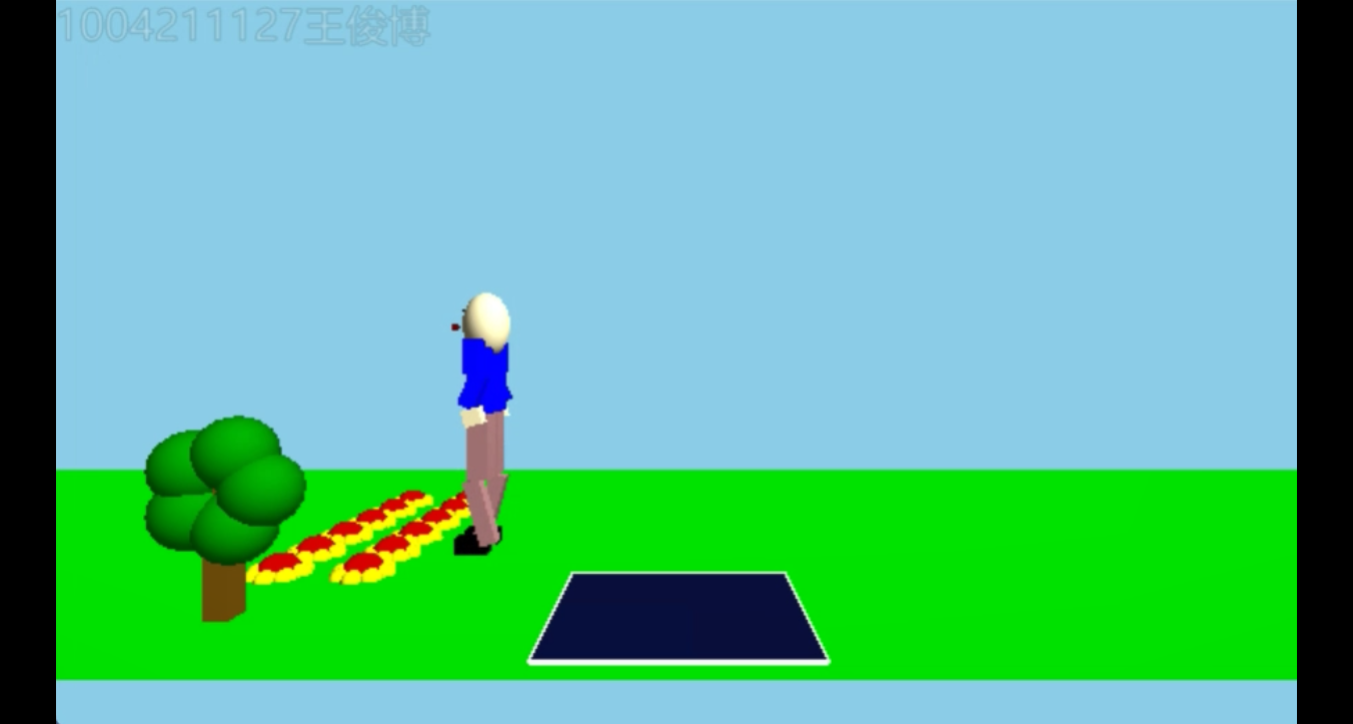




飞行



降落



1. 结果分析

实验结果表明，题目1~3实现较为完整，所需动画全部实现。题目4自由发挥部分，实现了室外的风景，由一颗树，一个花丛和一个泳池构成。机器人在室外飞行跨越泳池并完成降落。

1. 结论与展望
2. 心得体会

在构建场景时，应当注意准确计算每一个对象的位置，以便于高效通过系统的相似度检测，也能够使得场景更加精致、美观。

在实现动画时，要先设计好移动的顺序，仔细思考每个动作之间的关系，以此来设计和定义状态变量，才能在后期的开发过程中不造成混淆和逻辑错误。

实现机器人移动时，要注意实现关节的运动，保证真实性。

对待线上作业和线下作业都要高度认真，否则就会难以找到错误所在，带来不必要的麻烦。

1. 程序优缺点

每一个动作都对应有一个状态变量，如此设计逻辑清晰，易于理解，在个人和小型团队开发时有优势；缺点是状态变量可复用性差，在进行大型项目开发和大型团队协作时会影响工作效率。

动作的实现较为整齐连贯，整体的可观赏性较强，缺点是关节的转动较为生硬，不够真实。

室外场景布置使用了循环函数绘制花丛和树叶，优化了代码的整体结构，缺点是可复用性仍有局限。