

实验报告

课程名称：交互式媒体原理

开课学期：2023-2024学年第2学期

专 业：软件工程

年级班级：2022级01/02/03/04班

学生姓名：朱昊

学 号：222022321062008

指导教师：韩先锋

计算机与信息科学学院 软件学院

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目名称 | | 影随飞--姿态驱动的无人机姿态控制和目标追随系统 | | | |
| 实验时间 | |  | | 实验类型 | □验证性□设计性□综合性 |
| 一、实验目的  第三次实验报告旨在开发一个音效反馈系统，集成到无人机控制系统中，以增强用户与无人机交互的直观性和互动体验。通过使用PySide2工具，实现一个音效反馈类，对用户的操作进行实时的音频反馈。   1. 实验要求  利用PySide2实现了一个音效反馈类  1. 实验内容与设计   音效作为一种直接且易于实现的反馈方式，能够让用户更加直观地感知到无人机的当前状态和响应。PySide2作为一个跨平台的Python GUI工具集，提供了丰富的音频处理功能，适合用来开发音效反馈系统。  1,****环境搭建：**** 安装Python环境和PySide2库，准备音效素材。    ****2.音效反馈类设计：**** 设计音效反馈类的架构，包括音效加载、播放控制和音量调节等方法。   1. ****3.音效反馈类实现：**** 成功实现了一个音效反馈类，包括音效的加载、播放、停止和音量调节等功能。代码如下： class DroneAudioController():  def \_\_init\_\_(self):  super(DroneAudioController, self).\_\_init\_\_()  self.takeoff\_sound =QSoundEffect()  self.takeoff\_sound.setSource(QUrl.fromLocalFile('起飞.wav'))  self.takeoff\_sound.setVolume(1.0)  self.landing\_sound = QSoundEffect()  self.landing\_sound.setSource(QUrl.fromLocalFile('降落.wav'))  self.landing\_sound.setVolume(1.0)  self.picture\_sound = QSoundEffect()  self.picture\_sound.setSource(QUrl.fromLocalFile('拍照.wav'))  self.picture\_sound.setVolume(1.0)  self.flight\_forward\_sound = QSoundEffect()  self.flight\_forward\_sound.setSource(QUrl.fromLocalFile('向前飞行.wav'))  self.flight\_forward\_sound.setVolume(1.0)  self.flight\_backward\_sound = QSoundEffect()  self.flight\_backward\_sound.setSource(QUrl.fromLocalFile('向后飞行.wav'))  self.flight\_backward\_sound.setVolume(1.0)  self.fly\_to\_left\_sound = QSoundEffect()  self.fly\_to\_left\_sound.setSource(QUrl.fromLocalFile('向左飞行.wav'))  self.fly\_to\_left\_sound.setVolume(1.0)  self.fly\_to\_right\_sound = QSoundEffect()  self.fly\_to\_right\_sound.setSource(QUrl.fromLocalFile('向右飞行.wav'))  self.fly\_to\_right\_sound.setVolume(1.0)  self.keep\_distance\_sound = QSoundEffect()  self.keep\_distance\_sound.setSource(QUrl.fromLocalFile('距离锁定on.wav'))  self.keep\_distance\_sound.setVolume(1.0)  self.remove\_distance\_sound = QSoundEffect()  self.remove\_distance\_sound.setSource(QUrl.fromLocalFile('距离锁定off.wav'))  self.remove\_distance\_sound.setVolume(1.0) self.status = {"takeoff":False,  "landing":False,  "picture":False,  "flight\_forward":False,  "flight\_backward":False,  "fly\_to\_left":False,  "fly\_to\_right": False,  "keep\_distance":False,  "remove\_distance":False}def set\_drone\_status(self,status\_name):  if self.status[status\_name]==False:  self.status[status\_name]=True  self.play\_sound\_based\_on\_status(status\_name)  elif self.status[status\_name]==True:  self.status[status\_name]=False  def play\_sound\_based\_on\_status(self,status\_name):  sound\_mapping = {  "takeoff": self.takeoff\_sound,  "landing": self.landing\_sound,  "picture": self.picture\_sound,  "flight\_forward": self.flight\_forward\_sound,  "flight\_backward": self.flight\_backward\_sound,  "fly\_to\_left": self.fly\_to\_left\_sound,  "fly\_to\_right": self.fly\_to\_right\_sound,  "keep\_distance": self.keep\_distance\_sound,  "remove\_distance": self.remove\_distance\_sound  } if status\_name in sound\_mapping:sound = sound\_mapping[status\_name]print(121241)  sound.play()  *#time.sleep(1)* print(666)  self.set\_drone\_status(status\_name) if \_\_name\_\_ == '\_\_main\_\_':  app = QApplication(sys.argv)  controller = DroneAudioController()  controller.set\_drone\_status("remove\_distance")  print(controller.status["picture"])  sys.exit(app.exec\_())   *4.*****测试反馈：**** 在测试中，音效反馈系统表现出良好的稳定性和响应速度，用户对音效的实时性和多样性表示满意。  5.****系统集成：**** 音效反馈类与无人机控制系统的集成顺利，能够根据用户的操作和无人机的状态播放相应的音效。（交由其他组员完成） | | | | | |
| 四、实验结果 （可给出截图等加以说明）  实验结果表明，利用PySide2实现的音效反馈类能够满足无人机控制系统的需求，为用户提供了直观的音频反馈，增强了人机交互的体验。通过用户测试，验证了音效反馈系统的有效性和用户满意度。  集成到系统后，点击对应按钮以及无人机做出相关动作都会有提示音给用户反馈 | | | | | |
| 1. 结果分析及总结（对实验的结果是否达到预期进行分析，总结实验的收获和存在的问题等）   总结：本实验通过利用PySide2工具，成功开发了一个音效反馈系统，并将其集成到无人机控制系统中。实验结果表明，音效反馈系统能够提供实时、直观的音频反馈，显著提升了用户体验。通过用户测试，收集了宝贵的反馈信息，为系统的进一步优化提供了指导。未来工作中，我们将继续改进音效反馈系统，增加音效素材的多样性，优化音量控制和同步性，以满足更多用户的需求。  在实验过程中，我们发现了一些需要改进的地方：  ****音效素材的多样性：**** 用户对音效的类型和风格有不同的偏好，需要提供更多样化的音效素材供用户选择。  ****音量控制的智能化：**** 在不同环境下，用户对音量的需求不同，可以考虑实现音量的自动调节功能。  ****音效与操作的同步性：**** 部分用户反映在快速操作时，音效的同步性有待提高，需要进一步优化系统的响应速度。 | | | | | |
| 教  师  评  阅 | 实验成绩（A-E） | |  | | |
| 评语意见： | | | | |