Assignment 4

Introduction to Augmented Reality (EN.601.654 and EN.601.454)

Instructors: Dr. Adnan Munawar and Dr. Peter Kazanzides

TA: Haochen Wei

CA: Nick Greene

Due Date: Oct 14th 2025

Due Time: Midnight (12 AM) EST

Submission instructions:

a. Please show all your work. You are strongly encouraged to solve non-programming questions on paper
b. For programming questions **(Q2 Part e, and f, and Q3)**, instead of pasting your code in the assignment, create a **private** GitHub repository in your account called **<username>_ ar_2025_assignment4** where <username> could be either your GitHub ID or JHU ID. In this repo, **create separate folders and subfolders for each programming problem and its subparts** and name them appropriately to upload relevant code.
FOR Q3 YOU MUST USE AN APPROPRIATE UNITY .gitignore FILE SUCH AS THIS.
Add the instructors and TAs as collaborators to your **private** repository using our Github IDs (adnanmunawar, pkazanzides, robinwei99, nickgreene). This will enable us to view and grade your code privately
c. For **Q3** submit a screen recording which demonstrates all of the features alongside your submission
d. Convert your solution to a **SINGLE PDF** and upload on Gradescope via Canvas

Q1. **[5 Points]** Hardware Devices and Components for AR

a. [1 Point] In your own words explain what is the Vergence Accommodation Conflict (VAC)?

b. [1 Point] Explain in your own words how the Meta Quest 3s HMD tracks the hand controllers (both when they are visible and when they are intermittently occluded)

c. [1 Point] You are developing your own AR HMD. You decided to use Pancake lenses due to their advantages over Fresnel lens. What other component(s) in the HMD will be required/preferred due to your choice of using Pancake lenses?

d. [1 Point] You are developing your own AR HMD. If “cost” and “manufacturability” of the lenses were **not** the determining factors, why might you still choose Fresnel lenses over Pancake lenses?

e. [1 Point] You are developing the latest and greatest untethered AR HMD (with onboard processing and power) and cost is not a consideration. Why might you still pick a computationally less powerful processor over the most powerful processor out there, assuming they both fit inside the HMD and are easily available?

Q2. **[7 Points]** Kalman Filter for Pose Estimation

a. [0.5 Points] Write out the symbolic Kalman Filter equations
b. [1 Point] Explain in your own words the matrices A, B, P, Q, R *and* K
c. [0.5 Points] What is the difference between Kalman Filter and Extended Kalman Filter?
d. [1 Point] How does the Kalman filter (or EKF) help with pose estimation in AR?
e. [2 Points] Consider this **incomplete** Python example of a 1D Kalman Filter with position and velocity as state variables:
   https://github.com/JHU-AR-2025/Assignment-4/tree/main
   (You must be part of the **JHU-AR-2025** GitHub organization to view this repository)
   This example uses random number generators with Normal Distribution to emulate process and measurement noise. Complete the missing code (lines with question marks “???” based on Q2.a and what we learned in class). Show the output plot
f. [2 Point] Update the code from Q2.e. to **not** consider “measurements” in our prediction (correction) and show how the prediction evolves over time. Show the output plot

Q3. **[7 points]** Unity Development

Recreate the in-class demo game and add the new features mentioned below. You may reference the lecture recording, or Unity’s Roll-a-ball tutorial. Some of the new features will require you to read the Unity documentation or learn from tutorials online. **Please**

refer to the posted example video which demonstrates the required features. You MUST use an appropriate Unity .gitignore file in your submission!

Features from class session:

1. [1 Point] Game environment with a floor, walls, and a keyboard-controlled player sphere
   a. Camera follows the player using the initial offset
2. [1 Point] Collectibles disappear on contact and an on-screen text counter updates

New features:

1. [1 Point] A Start button; gravity is disabled, and the player cannot move until it is pressed
2. [1 Point] Jump on keypress (spacebar is mapped to the "Jump" action by default)
   a. Place several cubes in the air so jumping is required to collect them
3. [1 Point] A power-up capsule appears after all cubes are collected
   a. On touching the power-up: change the player's color and increase speed
   b. On-screen text indicates when the power-up is active
4. [1 Point] A special wall which can only be "broken" after the power up is active
5. [1 Point] A transparent goal zone on a different stage with a gap. The player must jump the gap. On touching the goal zone, disable the player and display "You win!"

Submission Instructions: Upload a screen recording of Play mode (no build required) to Gradescope. Create a private GitHub repo as outlined at the beginning of the assignment.

YOU MUST USE A UNITY .gitignore FILE SUCH AS THIS:
https://github.com/github/gitignore/blob/main/Unity.gitignore

Q4. **[1 Point]** Make your **Assignment 3** GitHub code public and transfer it to the JHU-AR-2025 GitHub Organization (https://github.com/JHU-AR-2025). Yes, this is the previous assignment. Make sure to include a **README.md** in your repository indicating your name for attribution, and elaborating on what the repository contains and how to run it.