Introduction

The purpose of this homework is:

1. To introduce you to the state-of-the-art in assistive HRI
2. To try out designing accessible technologies for yourself

All written answers should be in a single file in PDF format. All code should be submitted in a zip file with your PDF. This assignment assumes you are programming in Python; it may be possible to complete many assignments in C++ as well, but please get in touch with the professor or TA if you would like to take that option so that we can be sure you know what you're getting into.

Part 1: The Latest and Greatest in Assistive Technology

Choose a paper on assistive HRI for a specific population of users with disabilities from a human-robot interaction conference: HRI, Ro-Man, CHI, IVA, CSCW, ICMI

Choose a paper addressing the same population from the most recent ASSETS conference.

Choose a paper addressing the same population (OR using the same computational technique as one of the above two papers) from a technology-focused conference: ICRA, AAAI, IJCAI, RSS, IROS, AAMAS, CVPR, NeurIPS, CoRL

Use the advanced search function from IEEEXplore (https://ieeexplore.ieee.org/search/advanced) or the ACM Digital Library (https://dl.acm.org/search/advanced -- make sure you select "The ACM Guide to Computing Literature"), or use Google Scholar, putting the full name of the conference in quotes. If you'd like to read a paper from a conference not listed above, please post to the class discussion board or stop by homework help hours and the professor or the TA will let you know if it qualifies.

1. [2 points] Provide a link to the three papers. Summarize each of the papers in your own words (do not just copy/paste the abstract or introduction!) What population is being addressed? How does the system or algorithm provide assistance to users? How well does it work? What evidence is provided that it works?
2. [3 points] Compare and contrast the three papers: How do the introductions compare? What is different about the focus of the papers? How do they talk about end users? How do they talk about the technology? What kind of evidence do they provide that their system or algorithm works?

Part 2: Computational Tools

You'll need to have a working copy of linux, preferably Ubuntu 20.04. If you're running Windows or MacOS, you can find instructions for installing Ubuntu as a virtual machine in VirtualBox here:

https://linuxhint.com/install_ubuntu_virtualbox_2004/

If your computer isn't beefy enough to handle running virtual machines, please get in touch with the professor or TA to discuss your options.

0. Install Ubuntu 20.04 and the python3-tk package (`sudo apt-get install python3-tk`).

1. [1.5 points] Take a look at the documentation for the python turtle module

https://docs.python.org/3.3/library/turtle.html?highlight=turtle

Write a script that lets you drive the turtle around with the arrow keys. Add a red turtle to the scene to act as a goal. Add a player turtle in green that is controlled by the user using the arrow keys. Whenever the player turtle reaches the goal turtle, teleport the goal to a new location. Submit your code for this part.

2. [2.5 points] Add the ability to describe the scene: use a system call with the "say" or "spd-say" commands (on MacOS or Linux), or use pyttsx (https://pyttsx.readthedocs.io (Links to an external site.)) to enable your computer to tell a user verbally about the locations of the turtles. You can also use other sounds if you'd like.

A. What information does the system need to convey to make it possible to reach the goal turtle? Try out your system without looking at the display; what did you notice when testing?

B. Iterate on your design until you have a system that lets you consistently reach the goal turtle. How did you change things? Submit your code for this part.

3. [1 point] What lessons can we draw from this exercise about designing accessible interfaces?

Standalone homework questions

1. One of the principles of universal design is that information should be communicated in several ways. Pick a mood, an aspect of the robot's state, or some information about the world a user might find useful and write a program that lets your robot express that information in at least three different ways. Make a video of your robot and post it to either canvas or campuswire (or both). Upload your code & video in a zip file with your text answers.
2. Using the python turtle library from assignments 1 and 2, implement driving the turtle using the following interfaces, where "driving" means being able to adjust both the speed and heading of the turtle, and get it to follow any path you'd want it to (for example, a circle, triangle, and square) or go to and stop at a location using each of the following interfaces. For each part, provide your code, a description of your solution, and a reflection on how well it worked.:
    A. the mouse
    B. the arrow keys
    C. the "z" key on the keyboard (and no other keys or mouse)
3. Choose one of the principles of universal design for disability other than "perceptible information" and incorporate it into your turtle control game from part 1. What principle did you choose and how did you implement it? How well do you think it worked? What would it take to make the turtle game fully "universally designed"?