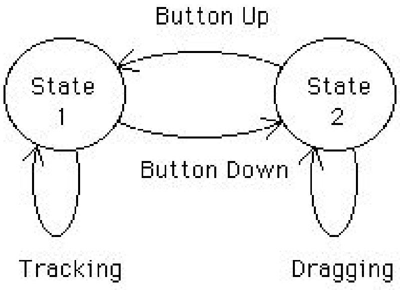
CSCI 130 – Input

Release Date: November 20, 2018
Due Date: December 6 , 2018 at 12:00PM

Overview

You will exercise creativity and logic to design the input controls for a car! First, for a deeper understanding of input and state models, you'll examine the car's main driving controls. Next, you will detail a state model protocol for an interactive task with a voice interface. Then, you will create a prototype using Actions on Google with Dialogflow and test it out with a user before finally creating a presentational video to creatively demonstrate your voice UI.

Part 1: Warm Up to State Models



A state model is a diagram that comprehensively describes the behavior of an input device: the possible device states and the transitions between them. The example on the right is a state diagram of basic mouse input. *Tracking* and *dragging* are states; *button up* and *button down* are transitions. See the Buxton Input reading and Pointing lecture notes for more examples.

You will be creating a state model for an automatic transmission car (if you haven't learned to drive yet, this is your chance!). Your state model will be more complex than the example above, as you notice there are a number of transitions and states that are rarely used but need to be considered when designing input devices. **The input controls you will handle will be 1) the gas pedal, 2) the brake pedal, 3) the gear selector. You only need to deal with 3 gears for this assignment: Park, Reverse, and Drive.** Assume that the car has been started and the engine is running. If you are not sure what happens in some scenarios, find online tutorials or ask friends, family, or TAs who drive, and cite that in your handin.

Make a state model to represent the different states and the transitions between them. Your state model should be one connected model, not disjointed smaller state models for each gear. Although we will accept hand-drawn diagrams if they look tidy and portfolio-ready (refer to style guide), we **recommend** using design software, presentation software like Powerpoint or Keynote, or flowchart software such as draw.io for a polished, easily editable diagram.

Part 2: Designing Input for Voice Interaction

Having gained some experience creating state models, you will now apply your skills to design for voice interaction! Applications like Siri, Alexa, Cortana, Google Now or even automated phone menus are staples of computer/phone interaction. Yet design-wise, voice interaction software is still an underdeveloped space, especially to do tasks while driving. You will dive into this problem space, creating and testing a state model for a conversational voice interface inside a car.

1. **Choose one of the tasks below to design. This will be your intent.** An intent is the underlying goal or task you user wants to accomplish.

Your voice interaction software will help someone fulfill their intent to **(choose one)**…

- Select a place enroute (gas station, grocery store, police station) based on some search criteria and get directions to navigate there, including dealing with detours
- **OR** place an online order for delivery to your destination, including entering payment info
- **OR** scout a location for an event, while creating that event with date, location, and finally sharing with the guests
- **OR** plan a recipe for when arriving home: search the Internet for ideas based on the recently purchased groceries in the car, then ask for step-by-step instructions
- **OR** another task of your choosing that is approved by the TAs, as long as it is complex enough! See this video by a Google speaker for inspiration. Post on Piazza (Private) to approve your idea before you start.

2. **Create a state model** to reflect your software's full range of interactions. Think through how fine-grained your states should be; will each individual word be a state? Every sentence? What makes sense for your interface? **See the bottom of this handout for an example state model.**

   Your state model **should *not* just focus on the "happy path,"** where the computer executes the user's commands perfectly on the first attempt. It should be a **flexible, conversational** interface that gracefully handles error states. For instance, what happens if the computer misunderstands the command? If it misunderstands only half of the command, or if the person makes a mistake halfway through? How does the computer know that two different phrases might have the same meaning? What if the person interrupts the computer? Just like you did for the previous state model don't forget to make one large connected state model rather than multiple smaller ones.

   We don't expect you to do extensive research into the current capabilities of voice recognition technology. We **do** expect you to keep the above questions in mind, consider how they apply to the task you are tackling, and incorporate them into your state model. See the last page of this handout for an example of a state model with error states. (The example task is much simpler than the task you will be working on, so expect your state model to be considerably more complex. You can turn in PDFs that are larger than US letter size if needed! Don't forget to test your document yourself before you submit it and make sure that the text is clear.) The TAs should be able to complete the task and know what happens if they make a mistake at any point or reach an error state.

3. **Create your task using Actions on Google.** You will be using Google Assistant's developer platform and editing it in Dialogflow. The link to the video below is helpful and will help you get started with this. This platform is used by companies to extend the services of their interface to be accessible through all Google Assistant devices but for the purpose of this assignment, we will be using this interface to build the state model you created in the previous step (Step 2) and then test it out in step 4. You can enhance your app by coding more advanced functionality in Dialogflow, but this is completely optional. You should be able to translate your task to Actions on Google without any coding. The links below should help you get started.

   Build Engaging Conversations for the Google Assistant using Dialogflow
   Handling Error States

   Remember, you will need to use a non-Brown or non-RISD gmail account to create this prototype. If you don't have one, you can ask a friend or make a new account. When you are done creating your prototype, go to the settings and **share it with the email cs130.input@gmail.com**. The TAs will be using your shared prototype to grade.

4. **Test your prototype with a user.** Find a user and test your prototype with them. This can be done by using the 'Try it Now' panel on the right side of the interface. Before you start, instruct your user on what their intent should be during the testing process. Then ask them to interact with the interface and try to fulfill the intent that you give them. **You will need to submit a video of the interaction between the user and the interface in addition to which, you should either include in the video or handout, the instructions you gave them.** Your video has to be of high resolution and can be taken from a phone or camera or a screen recording with audio of the user. The TA can see the screen and hear the user.

5. **Describe your conclusions from the testing session in one to two paragraphs**. What did you learn about your state model's strengths and shortcomings? Did anything confusing or unexpected happen? What did you learn from using the developer platform? Was there anything surprising you encountered? Did any of the results confuse you or intrigue you? Did it run exactly as expected. In your paragraph, try and reflect on the entire testing process.

6. **Update your state model diagram** based on the insights you gathered. You do not have to update your Google Developer Prototype, but if you do, make sure to include a note on your handout and write a short paragraph on what you updated.

7. Finally, **create a short commercial video** of an example scenario in your newly designed voice UI. Briefly introduce the idea and walk through a task from beginning to end. Try and advertise this to your audience. It could be a commercial for the hypothetical app for which you designed

the voice interaction or for a Voice Interaction device like Alexa, Siri, or the Google Assistant. Get creative here; take inspiration from this UX portfolio video or this Siri commercial and try to make your video portfolio-ready. Your potential users are your target audience, and you're trying to convince them why your product is great by demonstrating all the feature of your product. You may use materials from the testing session you did in part 3 and 4 (but not the whole video), as well as acted scenarios captured on your phone, animations, Powerpoint slides, clips edited with Adobe Premiere or iMovie, voice overs, or any other accompanying visuals you think might help present your work! Your video does not need to be long to explain the concept and demonstrate a voice interaction scenario; one minute is about right.

Handing in and Rubric

Submit a PDF with your three state models (one from Part 1, two from Part 2), text, and state model on the Google Interface, and links to your testing video and presentational video on YouTube/Vimeo. In addition, share your prototype with this gmail account. The pdf submission on Gradescope and email sharing the Google Home prototype is due by 12pm (noon) on Dec 6th.

Part 1 (3 points)

- 3 points — Comprehensive state model representing all 3 input controls, 3 gears, error states, transitions etc.

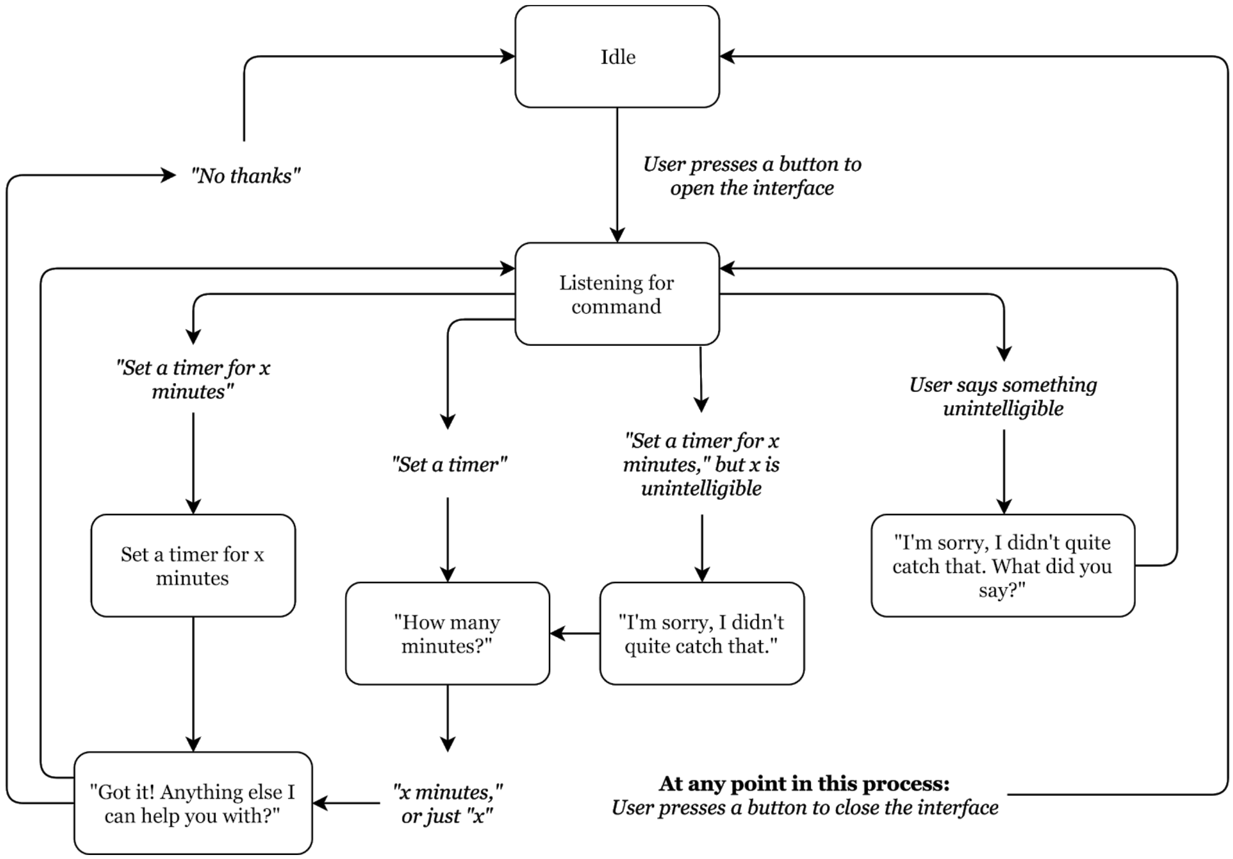
Part 2 (17 points)

- 5 points — Comprehensive state model representing the voice interface's possible actions and state, including idle and error states. A TA should be able to walk through your interface to complete the task and know what would happen if they make a mistake or the interface misinterprets their command.
- 3 points — Translating your state model onto the Google Assistant Developer.
- 1 points — Video recording of the user testing that shows the functionality of your prototype on Google Assistant Developer
- 2 points — Paragraph discussion of the test and insights on how to improve the state model
- 2 points — Updated state model based on your takeaways
- 4 points — Portfolio-ready presentational video of the idea and scenario for your interface
  - 2 points — follows an example task from beginning to end; shows full user journey and fully demonstrates the interface's capabilities
  - 1 points — is consistent with your state diagram conclusions from your user test
  - 1 point — creatively demonstrates your product in a compelling manner

Style (2 points)

- Check out the style guide on Canvas for more details!

SAMPLE STATE MODEL


Idle
User presses a button to open the interface
"No thanks"
Listening for command
"Set a timer for x minutes"
"Set a timer"
"Set a timer for x minutes," but x is unintelligible
User says something unintelligible
Set a timer for x minutes
"How many minutes?"
"I'm sorry, I didn't quite catch that."
"I'm sorry, I didn't quite catch that. What did you say?"
"Got it! Anything else I can help you with?"
"x minutes," or just "x"
At any point in this process:
User presses a button to close the interface