Programming: Feel the Power

A new, a vast, and a powerful language is developed for the future use of analysis, in which to wield its truths so that these may become of more speedy and accurate practical application for the purposes of mankind than the means hitherto in our possession have rendered possible.

--Ada Lovelace, 1842.

What is programming?

What is computer science?

“Computer science is the study of what computers [can] do; programming is the practice of making a computer do things”

-Random Guy on a discussion forum

What is a computer without programming?

What is a computer without programming?

- A computer without any programming is about as useful as a cup of coffee without the cup.
- *Hardware* needs *software* to do anything. The physical computer is the hardware, the code is the software
- When you buy your computer you already have tons of programs installed.
- Everything must be programmed!

What can computers do better than humans?

What can computers do better than humans?

- Add quickly, multiply quickly
- Use the Internet
- Render graphics for games/video
- Read text quickly
- Scraping the Web (what Google does)
- Processing Data

What can humans do better than computers?

What can humans do better than computers?

- Image recognition
- Understanding language (detecting sarcasm)
- Unstructured problem-solving (word problems)
- Acquiring and processing new information (learning)
- Physical tasks (http://www.youtube.com/watch?v=gy5g33S0Gzo)

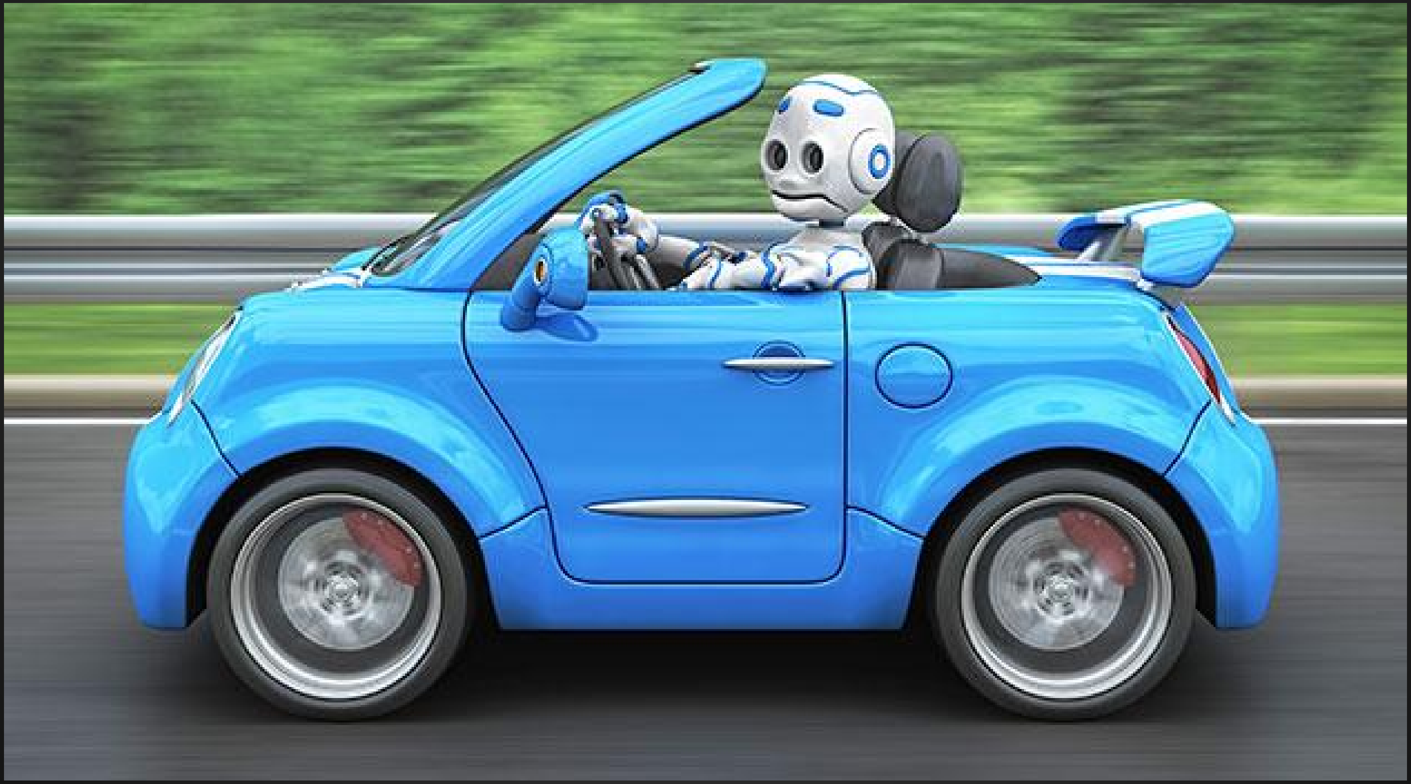
Let’s say you have a robot and you want it to drive your car

Writing a program is a lot like explaining to a dumb robot how to drive a car

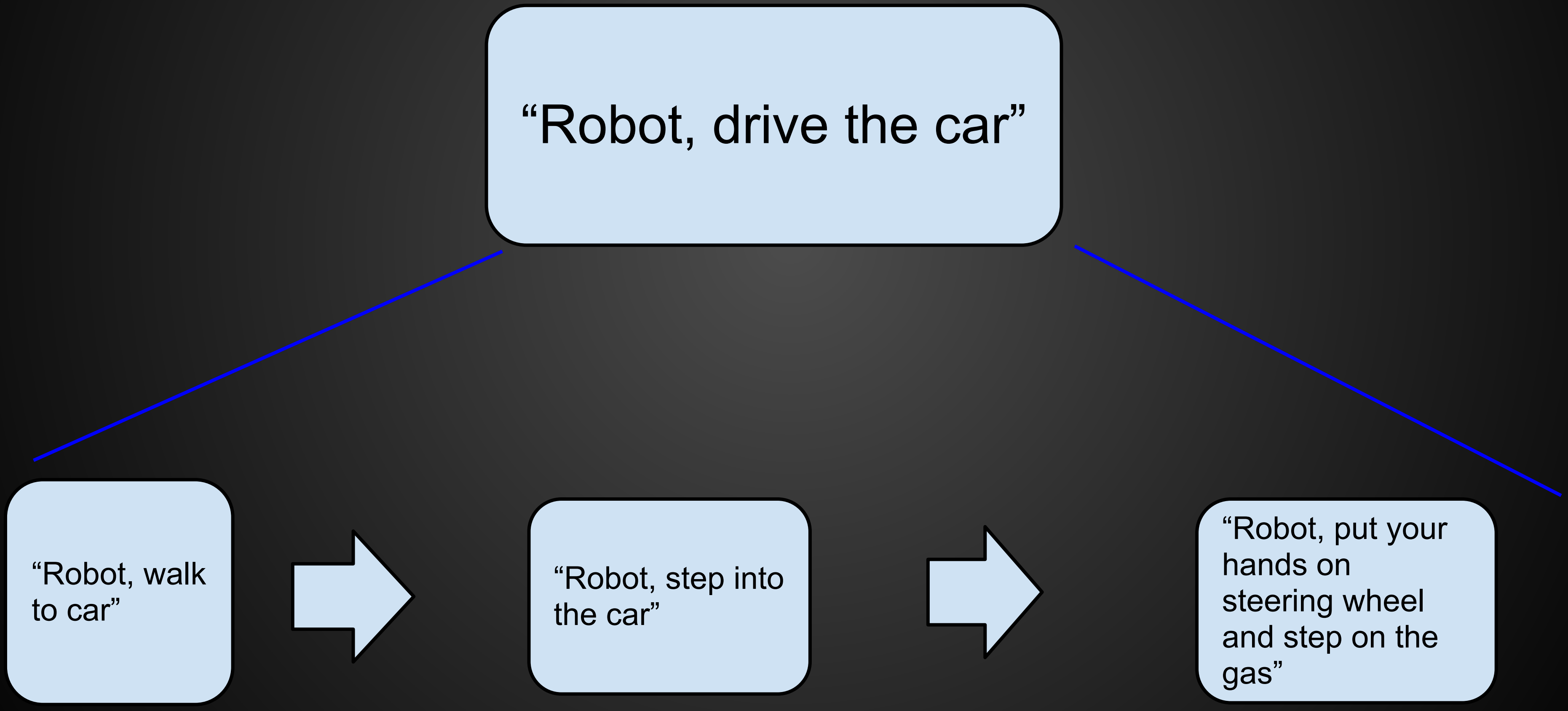
Think this:

Instead of this:

It’d be simple just to say “Robot, drive the car”



Break the task down into multiple smaller tasks that the robot can execute



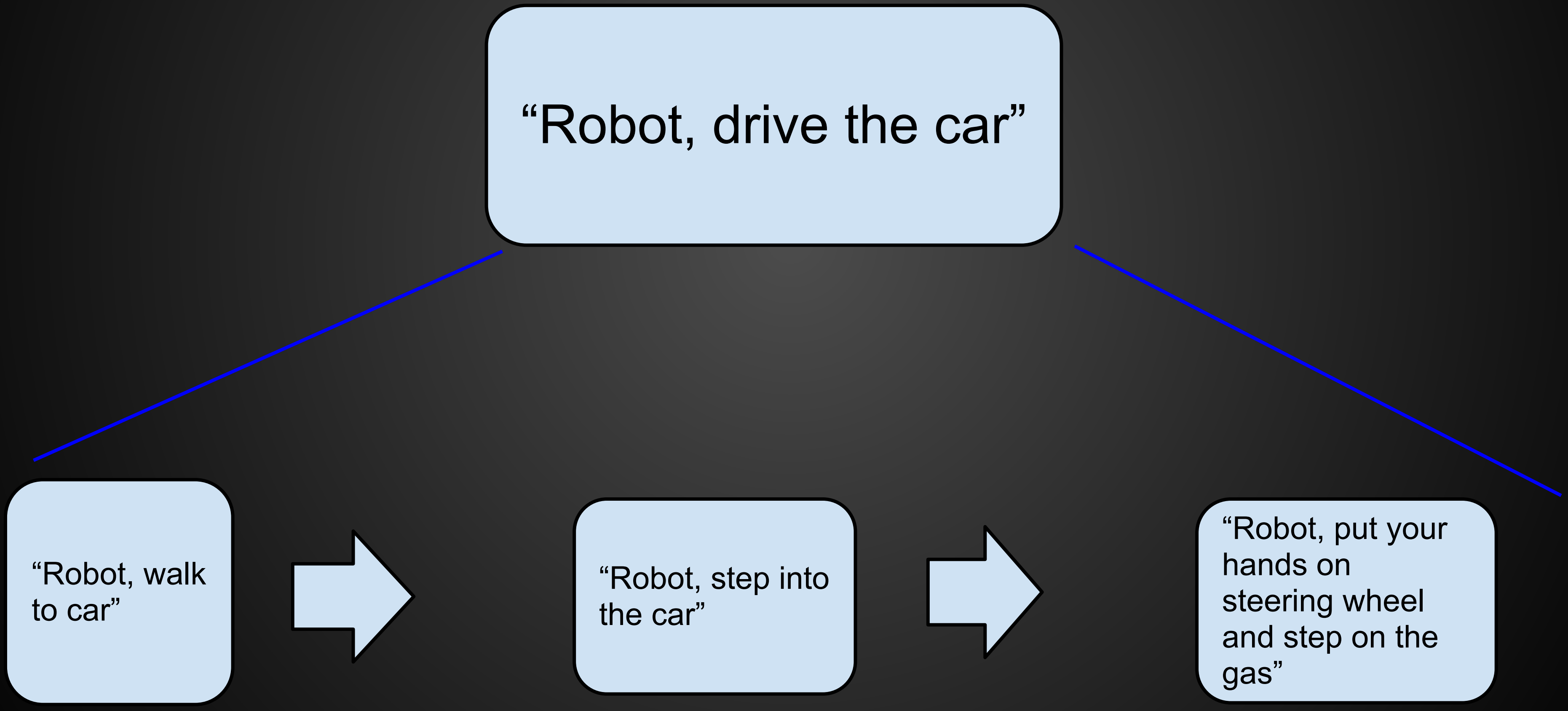
“Robot, drive the car”

“Robot, walk to car”

“Robot, step into the car”

“Robot, put your hands on steering wheel and step on the gas”

What are some problems with explaining it to the robot in this way?



“Robot, drive the car”

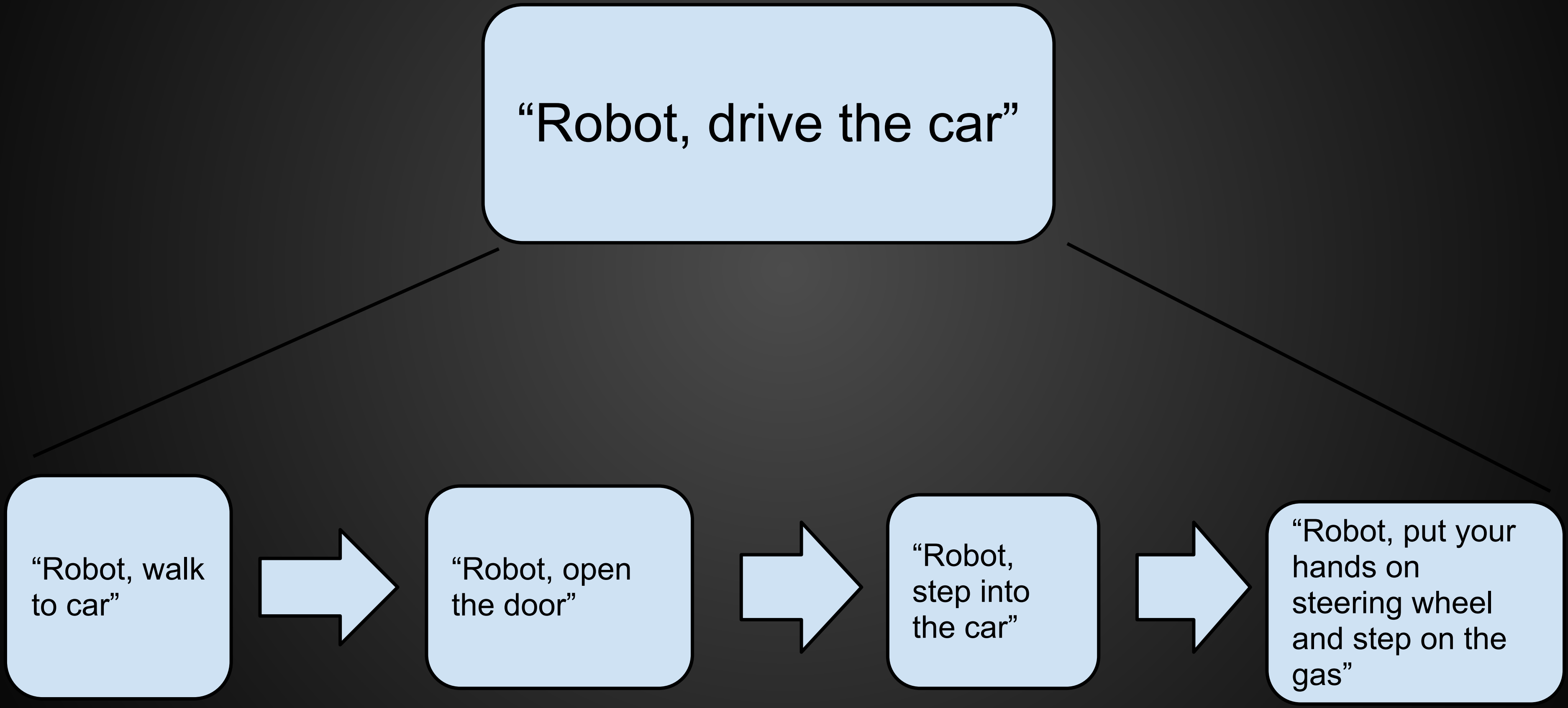
“Robot, walk to car”

“Robot, step into the car”

“Robot, put your hands on steering wheel and step on the gas”

Error 1: Robot should open the door before stepping in

Error 1: Robot should open the door before stepping in



“Robot, drive the car”

“Robot, walk to car”

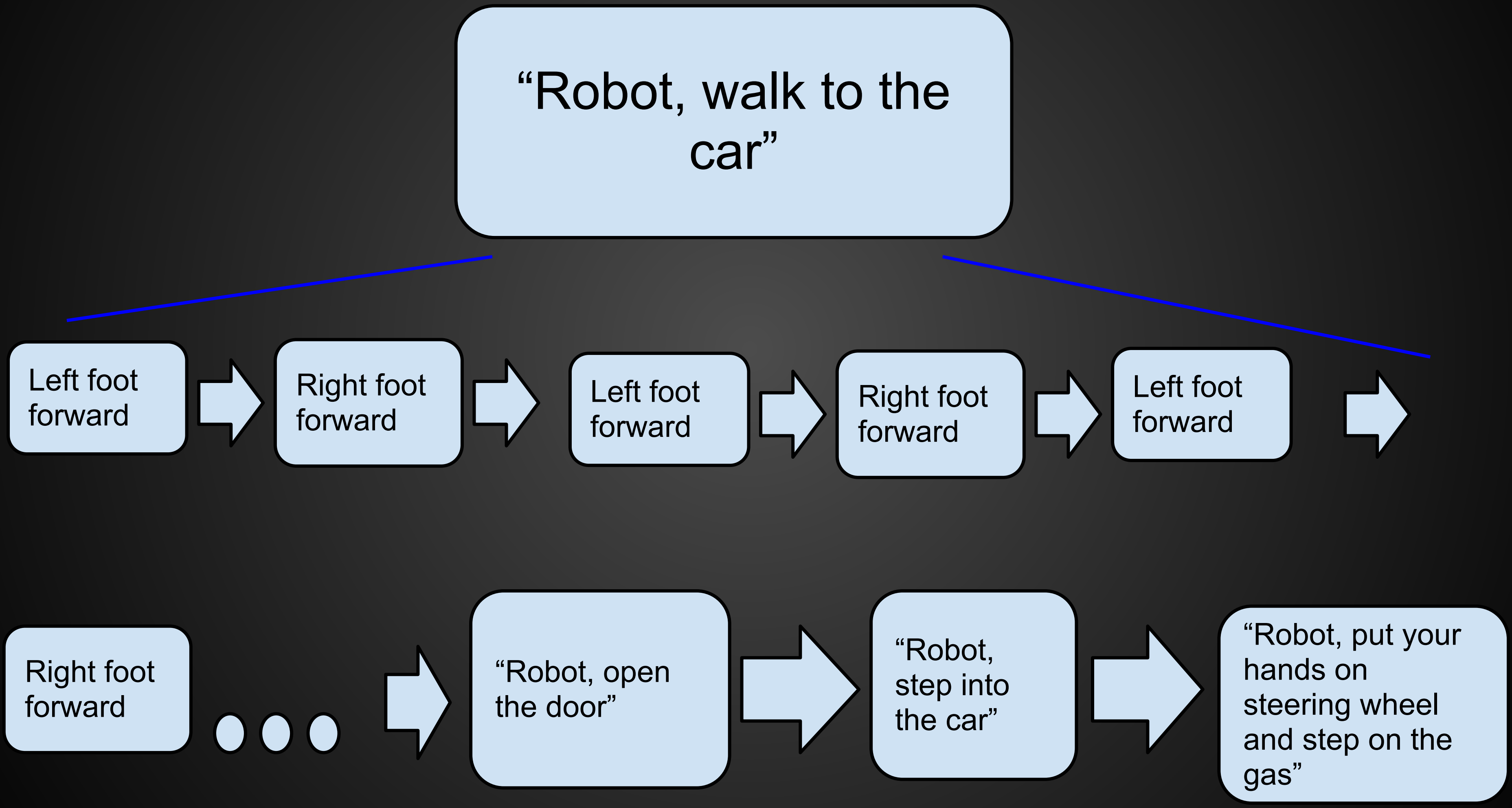
“Robot, open the door”

“Robot, step into the car”

“Robot, put your hands on steering wheel and step on the gas”

Error 2: Robot may not know how to walk

Error 2: Robot may not know how to walk



"Robot, walk to the car"

Left foot forward

Right foot forward

Left foot forward

Right foot forward

Left foot forward

Right foot forward

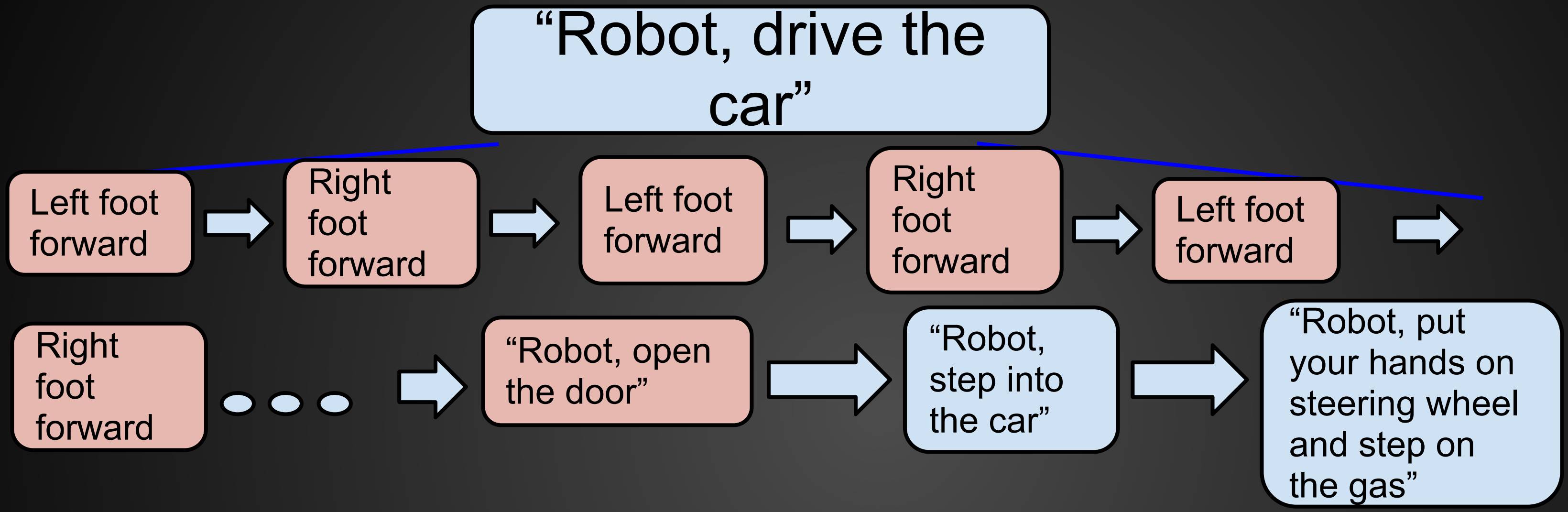
. . .

"Robot, open the door"

"Robot, step into the car"

"Robot, put your hands on steering wheel and step on the gas"

How does that translate into code?



“Robot, drive the car”

Left foot forward

Right foot forward

Left foot forward

Right foot forward

Left foot forward

Right foot forward

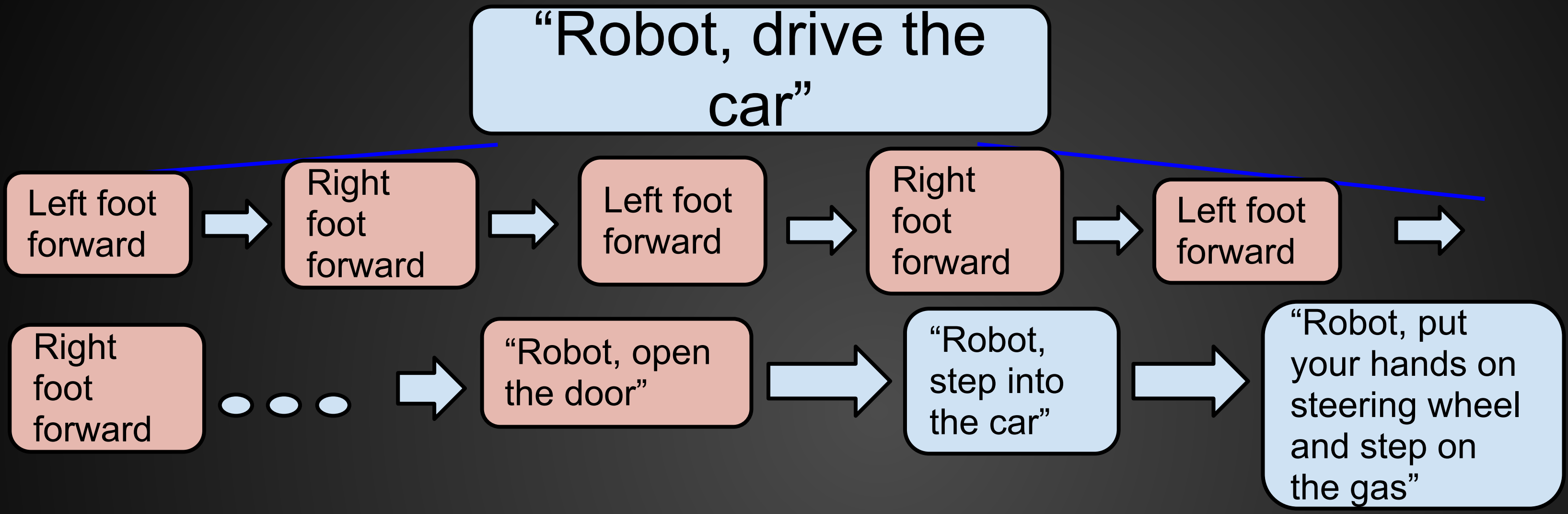
“Robot, open the door”

“Robot, step into the car”

“Robot, put your hands on steering wheel and step on the gas”

```
r = robot()
r.left_foot_forward()
r.right_foot_forward()
check_if_robot_is_at_car()
r.left_foot_forward()
r.right_foot_forward()
check_if_robot_is_at_car
r.left_foot_forward()
r.right_foot_forward()
check_if_robot_is_at_car()
r.left_foot_forward()
r.right_foot_forward()
… and so on…
check_if_robot_is_at_car()
r.open_car_door()
```

How does that translate into code?



“Robot, drive the car”

Left foot forward

Right foot forward

Left foot forward

Right foot forward

Left foot forward

Right foot forward

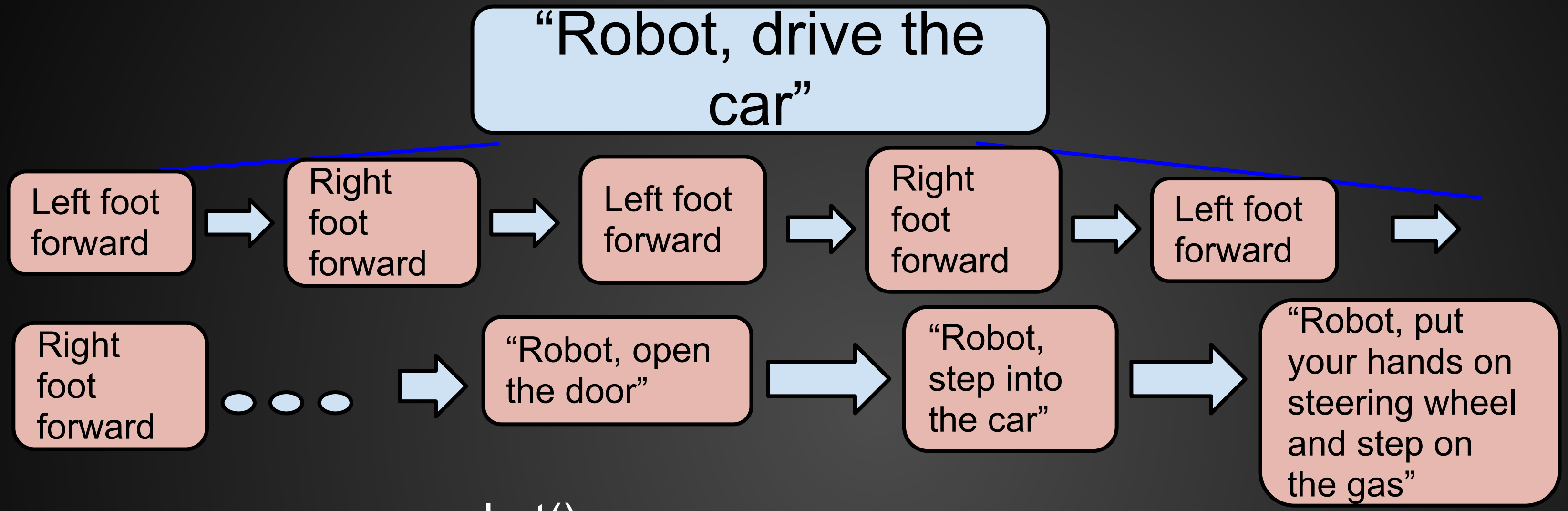
“Robot, open the door”

“Robot, step into the car”

“Robot, put your hands on steering wheel and step on the gas”

```
r = robot()
while (robot is not at car)
        r.left_foot_forward()
        r.right_foot_forward()
r.open_door()
```

How does that translate into code?



"Robot, drive the car"

Left foot forward

Right foot forward

Left foot forward

Right foot forward

Left foot forward

Right foot forward

"Robot, open the door"

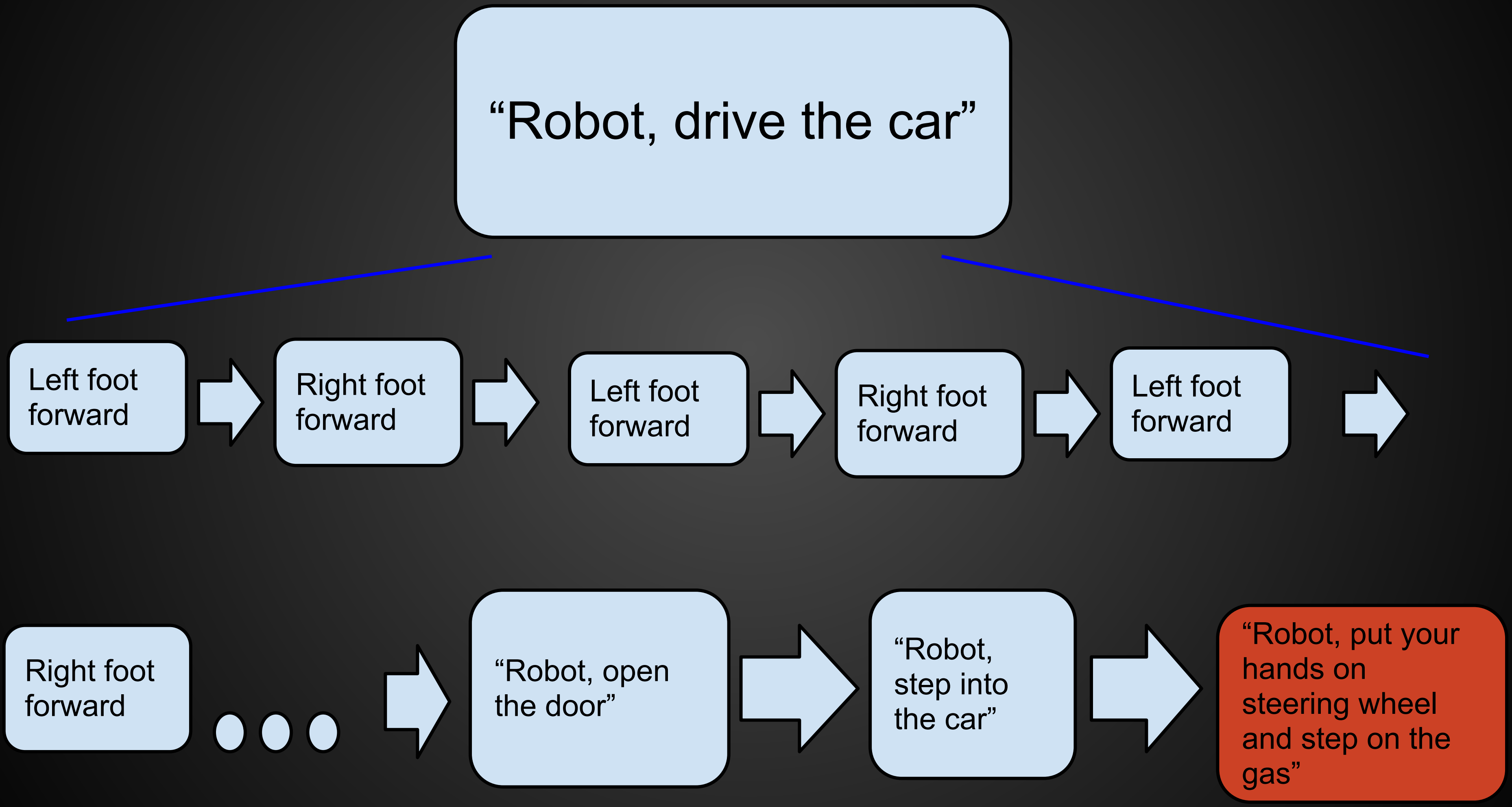
"Robot, step into the car"

"Robot, put your hands on steering wheel and step on the gas"

```
r = robot()
while (robot is not at car):
        r.left_foot_forward()
        r.right_foot_forward()
r.open_door()
r.step_into_car()
r.
place_hands_on_steering_w
heel()
r.step_on_gas()
```

Error 3: “Robot put your hand on the steering wheel and step on the gas” is a terrible idea

Error 3: “Robot put your hand on the steering wheel and step on the gas” is a terrible idea



“Robot, drive the car”

Left foot forward

Right foot forward

Left foot forward

Right foot forward

Left foot forward

Right foot forward

“Robot, open the door”

“Robot, step into the car”

“Robot, put your hands on steering wheel and step on the gas”

Error 3: “Robot put your hand on the steering wheel and step on the gas” is a terrible idea

- If approach stop sign, stop and look both ways before going.
- If a car is in front of you, slow down.
- If a you hit a red light, stop.
- Let’s say robot checks each of these conditions every 5 feet.

```
r = robot()
while (robot is not at car):
        r.left_foot_forward()
        r.right_foot_forward()
r.open_door()
r.step_into_car()
r.place_hands_on_steering_wheel()
r.move_forward_5_feet()
if (robot at a stop sign):
        stop, look both ways
if (car in front of robot):
        slow down
if (robot at red light):
        stop and wait until green
r.move_forward_5_feet()
if (robot at stop sign):
        stop and look both ways
… and so on ...
```

Lots of code...

```
r = robot()
while (robot is not at car):
        r.left_foot_forward()
        r.right_foot_forward()
r.open_door()
r.step_into_car()
r.place_hands_on_steering_wheel()
r.move_forward_5_feet()
if (robot at a stop sign):
        stop, look both ways
if (car in front of robot):
        slow down
if (robot at red light):
        stop and wait until green
r.move_forward_5_feet()
if (robot at stop sign):
        stop and look both ways
```

```
if (car in front of robot):
        slow down
if (robot at red light):
        stop and wait until green
r.move_forward_5_feet()
if (robot at stop sign):
        stop and look both ways
if (car in front of robot):
        slow down
if (robot at red light):
        stop and wait until green
r.move_forward_5_feet()
if (robot at stop sign):
        stop and look both ways
```

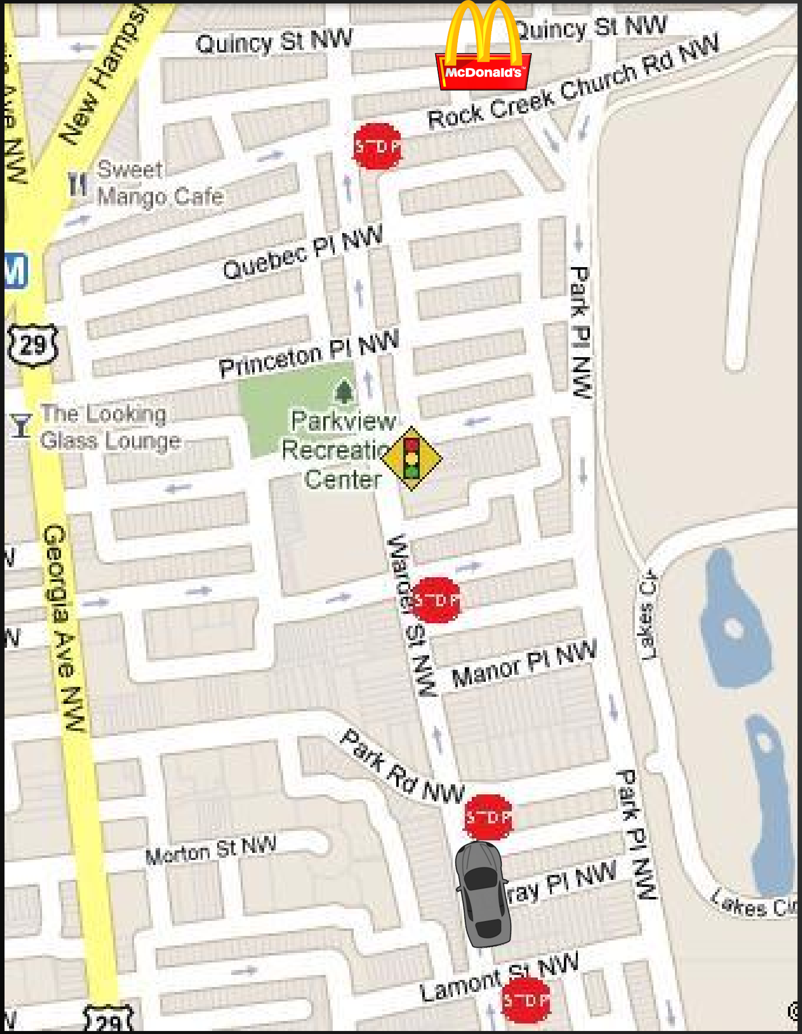
… and so on until robot stops driving ...

… this is a lot of repeating code ...

Simplified

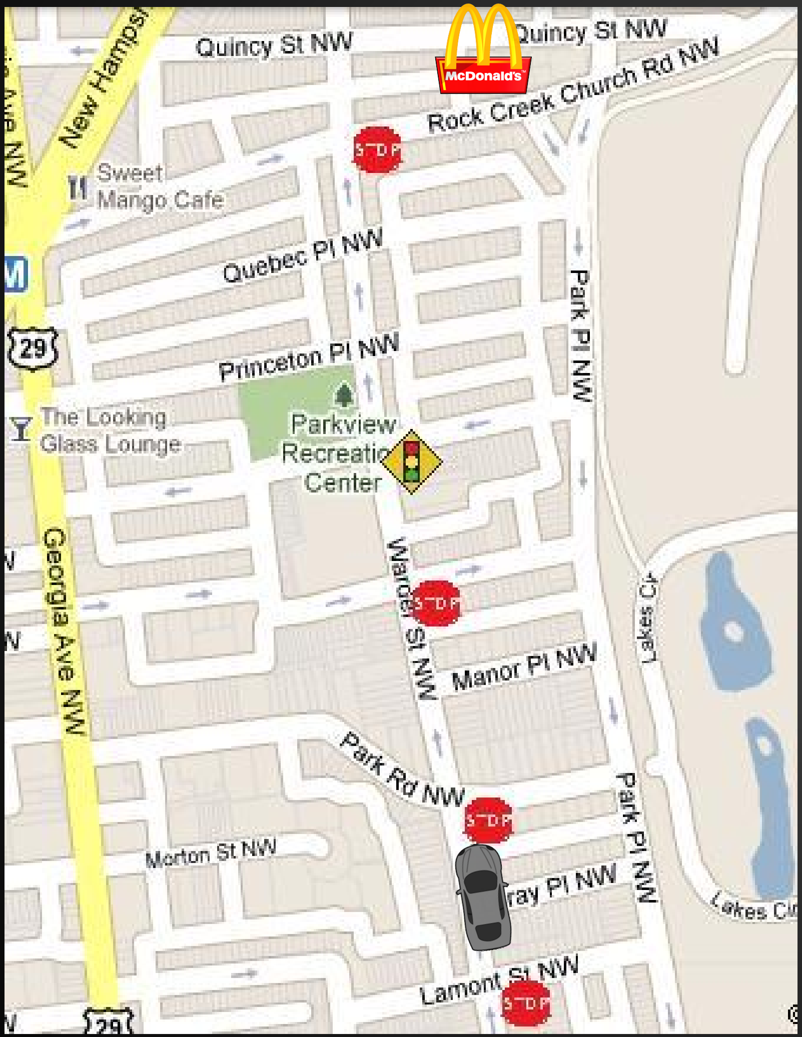
```
r = robot()
while (robot is not at car):
    r.left_foot_forward()
    r.right_foot_forward()
r.open_door()
r.step_into_car()
r.place_hands_on_steering_wheel()
while (robot is still driving):
    r.move_forward_5_feet()
    if (robot at a stop sign):
        stop, look both ways
    if (car in front of robot):
        slow down
    if (robot at red light):
        stop and wait until green
```

Challenge problem: Drive to McDonalds on Rock Creek Church Rd.



```
r = robot()
while (robot is not at car):
        r.left_foot_forward()
        r.right_foot_forward()
r.open_door()
r.step_into_car()
r.place_hands_on_steering_wheel()
while (robot is still driving):
        r.move_forward_5_feet()
        if (robot at a stop sign):
                stop, look both ways
        if (car in front of robot):
                slow down
        if (robot at red light):
                stop and wait until green
```

Challenge problem: Drive to McDonalds on Rock Creek Church Rd.



```
r = robot()
while (robot is not at car):
        r.left_foot_forward()
        r.right_foot_forward()
r.open_door()
r.step_into_car()
r.place_hands_on_steering_wheel()
while (robot is still driving):
        r.move_forward_5_feet()
        if (robot at a stop sign):
                stop, look both ways
        if (car in front of robot):
                slow down
        if (robot at red light):
                stop and wait until green
        if (robot at Rock Creek Church Rd.):
                turn right
        if (robot at Mcdonalds):
                turn left
```

Side note about these errors

- In programming, we call these errors *bugs*. We’ll talk more about bugs later.
- If your instructions don’t work, you spend a great deal of time *debugging*: looking through the execution of your instructions for bugs and fixing them.
- Sometimes there is a misunderstanding between what you want the computer/robot to do and what the computer/robot thinks you want it to do.

What is programming?

- Programming is POWERFUL
- Think of your computer as your personal assistant. You tell it to do things and it does it for you.
- Programming is the language you use to communicate with your computer.
- Computers are extremely literal so it is often a challenge to figure out how to get your point across!
- Thus, programming takes time. One important skill is deciding when it is worth writing a program to do something for you.

About the class/instructors

- 2nd time teaching
- Ben
- Emily
- Hurshal
- Jiann
- Lu