Lab 5: Wildlife Conservation

Wildlife Conservation with IoT

DISCLAIMER: This lab looks long - do not worry! Most of this document is more like reading material than a lab. Our goal is to provide you with extensive descriptions, to walk you through each step, so you will not get stuck, and to also teach you additional things. It is ok to skip over parts of the reading if you feel comfortable and don't feel like you need the extra help.

Among 25 pages, **only 4 pages are the description for the assignment**. 15 pages of them in step 2 and step 3 are app guidance added by our dev team in case anyone has issues using the app. Pictures also take most of the space. You can feel free to skip those pages if you don't run into problems. There are also 4 pages in step 1 to provide you with some fun background for animals. You don't have to go through it carefully, but we recommend you to read the whole thing. To save time, you can also get through them with just a few minutes if you follow the bolded words. **The actual assignment description is highlighted in yellow.** The rubric takes the last 2 pages.

Why should we care about wildlife?

> "Like us, animals feel love, joy, fear and pain, but they cannot grasp the spoken word. It is our obligation to speak on their behalf ensuring their well-being and lives are respected and protected."
> — **Sylvia Dolson**

The rate at which animal species are going extinct is rapidly increasing. Current estimates are that 10,000 species go extinct every year, halving the world's wildlife population just within the last 40 years. This rate of extinction is estimated to be 1000 to 10000 times higher than the "natural" extinction rate, and is widely believed to be caused by humans and effects they cause, such as deforestation, animal exploitation, overpopulation, and climate change. It is widely believed that humans are causing the 6th greatest extinction event on our planet[1], on par with extinction events such as the Cretaceous-Paleogene asteroid strike which wiped out the dinosaurs, and the Oxygen Catastrophe 2.45 billion years ago which led to the longest period of glaciation in earth's history known as "Snowball Earth".

[1] http://time.com/3035872/sixth-great-extinction/

Most people don't think about how important wildlife is. We don't think of where our food comes from, the countless vaccines, the oxygen we breathe. Our daily lives are so tightly coupled with other living things on the planet, in a highly complex intertwining that we are nowhere near understanding. When we read in the news about honeybees disappearing[2], which could someday prevent us from growing food, we don't think about how an estimated 90% of human diseases, including HIV, the flu, and Ebola, have origins from animals. When we were kids we all read books about animals, had stuffed animals. How would you feel, looking at your kid someday, and telling them there aren't bears anymore?

Because of that, some of the greatest heroes on our planet dedicate their lives to studying the behavior of wildlife. These studies help us conserve and protect our ecosystem. Protecting our ecosystem improves our food security, by improving nutrition, health, and dietary diversity of our agriculture. They help us improve our public health (over 90% of traditional medicines come from wild plants and animals). And they help protect our ecosystem for future generations to thrive. But more than that - the amazing and beautiful wildlife placed on this earth was not created by people; it was a priceless gift for us, and it is our responsibility to protect its diversity and safety.

What will you learn in this lab?

In this lab, you will learn several things:

- A real IoT application (animal conservation), which will give perspective on general IoT application development
- IoT data dissemination protocols, why they are important, and how they are implemented
- Practical aspects of IoT deployment, including sensor deployment, monitoring, and real-time data analysis of a live deployment

In this lab, **you will design, build, and deploy a real IoT application from the ground up**. You will take the role of a designer/operator working for a wildlife refuge in sub-saharan Africa. You are responsible for a 50 square mile territory. On your territory is a diverse range of animals, including zebras, lions, and elephants. Since this is a big area, you can't simply walk around and watch the animals. Instead, **you will perform tracking, where you affix IoT sensors to the animals, which monitor attributes about the animal such as its location, movement patterns and so on.**

Ok, we will now embark on our adventure! Sit down, buckle your seatbelts, and perform the following steps:

[2] https://www.sciencenews.org/article/mystery-vanishing-honeybees-still-not-definitively-solved

Step 1: Research The Problem

When you design an IoT device, your first step is to understand the problem you are addressing. What problem are you trying to solve? What does your device need to do exactly? What sort of environmental conditions will it be exposed to? Does it need to sense or actuate on its environment - if so, how specifically?

In this first step, you will practice the process of gaining understanding of an IoT problem domain. You will perform some basic research about the problem, which will assist you in thinking through how to design your device. As you read through this background material, try to think about what sort of behaviors you would want to monitor in these animals to help them, and how you could design a device to monitor those behaviors (what kinds of sensors you would want to use, how you would record/store data, etc).

Step 1a: Learn about Zebras

Figure 1: African zebras. Left to right: (a) close-up view of muzzle and mane (b) mother zebra with foal. Did you know that a group of zebras is called a "dazzle"?

In this section we talk a bit about zebras. We encourage you to read more[3].

The zebra is a strong and social animal that lives in the saharas of Africa. There are three main species: the plains zebra, Grevy's zebra and the mountain zebra. All three are alike in many ways, but their general stripe patterns and sizes are the main things that make them different from

[3] A good starting point is the wikipedia article for Zebras (https://en.wikipedia.org/wiki/Zebra), and citations therein.

each other. They also live in different areas of Africa. The plains zebra is the most common of the three zebras. They are medium-sized, have thick bodies and shorter legs. Also, they live in savannas and temperate grasslands (plain), which is the main area our research is based on.

Someone may ask: does the zebra have **black with white stripes or white with black stripes**? This is a quite controversial question. It is commonly believed that zebras have black stripes on their white bodies since the stripes gradually disappear as they approach the belly and inner thigh. But actually zebras have black skin underneath. Something without controversy is that each of their **strips are unique**, just like our human fingerprints. And this type of coat is well used as camouflage to confuse the predators (won't be able to know the exact number of zebras in the groups) and lower the attraction to bugs like the bloodsucking horseflies.

Zebras **value their family members and like to live in large groups**. A big herd of zebras could have from hundreds to thousands of individuals. It is formed by small family groups. The basic family group contains five to twenty individuals and serve different roles as stallion, mares and offsprings. They stay in the same family groups for many years, take care of and protect each other, even when they congregate into large herds. Their **sound and facial expression are their main language to communicate**. They could bark, bray, snort or huff to express themselves. Also sometimes they have wide-open eyes or bared teeth to express their feelings. They would form a stronger relationship bonding in the family group by social grooming. And When there's a danger, they would pull their ears back to alert other members. Fun fact about zebras is that they are **herbivore but not afraid of confronting their predators**. Running away is the usual tactic, but stallion will stand his ground while the rest of the family runs away in zigzag fashion and sometimes provide predator a defensive kick, which can cause serious injury to the predators like lion or hyena.

Zebras are **nomadic animals**. They don't have a specific territory. Because of the alternating wet and dry seasons, the river level varies throughout the year. During the dry season, water holes and rivers dry up, leading to water and food shortage and threatening grazing animals' survival. And this is when the incredible journey begins. Thousands of zebras immigrate to the south in a clockwise direction starting from early December in order to find greener grassland and more water supplies to provide adequate nutrition to mares and offsprings during that time. According to the observation of Robin Naidoo (a senior conservation scientist at the WWF), the herds could trek for a hundred and fifty miles within two weeks. They spend the next two months in the south enjoying the fruits of their hard work and head back to the north when the wet season's back there.

Step 1b: Learn about their predators

Figure 2: Zebra predators. Left to right: (a) Adult African male lion displaying mane (b) pride of lions observing prey. Lions have loud roars, and can be heard from 8km away!

One environmental factor that is quite important to Zebras are their predators. In this section we provide some background on a key predator of zebras: lions.

The name for lion in Swahili, an African language, is ‘simba’. African lions are mostly found in sub-Saharan Africa. They used to be found all over Africa but now they have mostly disappeared from North Africa and are at risk of extinction in West Africa. Lions live in **savannahs and grasslands, where some cover and plenty of water can be found**. African lions are the **most social** of all big cats and live together in prides (a social structure to describe groups of lions), which the size ranges from 1-21 females.About ⅔ of daughters are adopted into the existing pride, while the other ⅓ go and create their own prides. All sons leave the pride in search of an unrelated mate. Males are generally 1.5 times larger than females and every lion’s mane is different. Some do not even have a mane.

Lions generally occupy the same territory for generations and are **highly territorial**. Females will fight invading females while males fight invading males. Male lions use their roar to declare/warn about their territory. A lion’s roar can be heard up to eight kilometers away. Lions can use the roars to guess the number of males in another pride and will attack if they outnumber the other. Most often prides do not fight but warily gaze at one another and pass by.

Lion territory is smaller when there is more prey and larger when there is less prey. Roaring is done before a night hunt or when waking up at dawn. Lions urinate, defecate, or rub against grass to mark territory. Lions will eat any meat they find including carrion and will steal from hyenas and will hunt if they see an injured or sick prey regardless of when their last meal was. **Female lions are the main hunters**. While they’re out looking for food, the males guard the pride’s territory and their young. Despite this, the males eat first. While females usually hunt, males in smaller groups will also hunt. Lions suffer lots of injuries while hunting prey, some of which are debilitating.

90% of a lion's hunt will fail. They do not pay attention to wind direction which may alert prey due to smell. They also can only run short distances. They stalk prey from bushes and then leap out and bite the necks to suffocate their prey. In groups they surround prey from all sides to cause panic. Lions will eat till they are full and then will remain in the area of their kill for a couple of days. Usually, **hunting is done for 2-3 hours a day** and lions **spend the rest of the time lazing/resting**. When prey is scarce they can rest for up to a week without much activity.

Lions **bond to one another by rubbing their bodies against one another**. If unwarranted, the lion will be slapped. When lions open their mouths and "smile" they are actually smelling the air and communicating with one another. Prides do not stick together the whole time but actually scatter and rejoin often.

Step 1c: Read a research paper on IoT animal tracking (Extra Credit)

When you want to find out some information, you may go to the library, or go on Amazon, to find a reference book. You may go on Wikipedia or look in an encyclopedia. But long before information reaches these sources scholars study and debate over information to determine what is true. When scientists make a finding, they first publish their results in an academic conference (or journal, or workshop) to get their results peer-reviewed. Since IoT is such a new field, a lot of the current results haven't had time to find their way into books, so you may have to turn to academic publications to find out information.

In this section, we will get a little experience in reading an academic paper. Please click on this link to download the publication "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet". This is an example of what graduate students do when they go to grad school. Everyone thinks grad students just sit in an office all day but good ones go out and actually do stuff like this. This paper describes an academic study of a Zebra population and how IoT sensors were used to conduct the study. Please read the Title, Abstract, and Sections 1-3. Also skim over Section 4 but it is ok if you don't understand everything in that section, just get the basic ideas. This paper will give you some background on animal studies and relevant practical issues[4].

Step 2: Design Your Device

[4] A lot of people think they can't read research papers because they pick them up and they seem too hard or esoteric. But anyone can learn to do it actually, there are certain tricks to read them, it's kind of like reading Wikipedia, that's probably confusing too for people who see it the first time, but gaining experience with how papers are written can make them easier to read. For more thoughts on how to read research papers you can look at http://blizzard.cs.uwaterloo.ca/keshav/home/Papers/data/07/paper-reading.pdf .

Now that you've had a chance to think about your problem statement and challenges, the next step is to design your device. Since we will be experimenting with the device virtually it is important we construct it in a virtual space. We will be using the Illinois IoT Playground, a VR-based environment invented (no VR glasses needed for this lab) and developed here at the University of Illinois, which allows you to create devices, then interact with them in a virtualized environment.

Step 2a: Design the device

As a first step, take a blank piece of paper (or google doc etc), and start writing down a design that captures what you want to do (monitor the Zebra population). Probably you want to put some sort of sensor-enabled device on the animals so you can monitor it. How would you do it exactly? Think about:

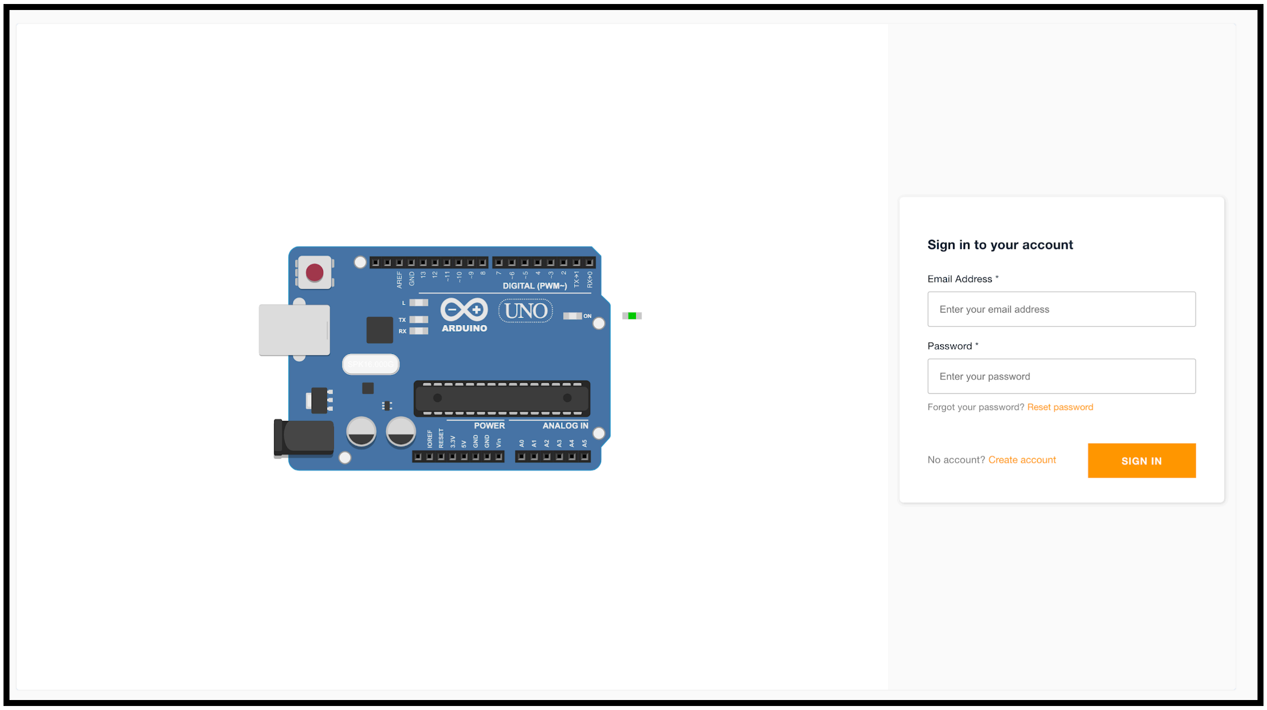
1. What sensors would you need
2. How would they be connected
3. What sort of code would you run on there
4. Where would you put the sensors on the animal
5. Physical construction (waterproof? weight? where would you put it on the animal?)

Just think it through. Take a picture of your piece of paper and include it in your report. Also include a discussion of your design considerations answering those 5 questions in your report.

Step 2b: Get familiar with the IoT Playground app

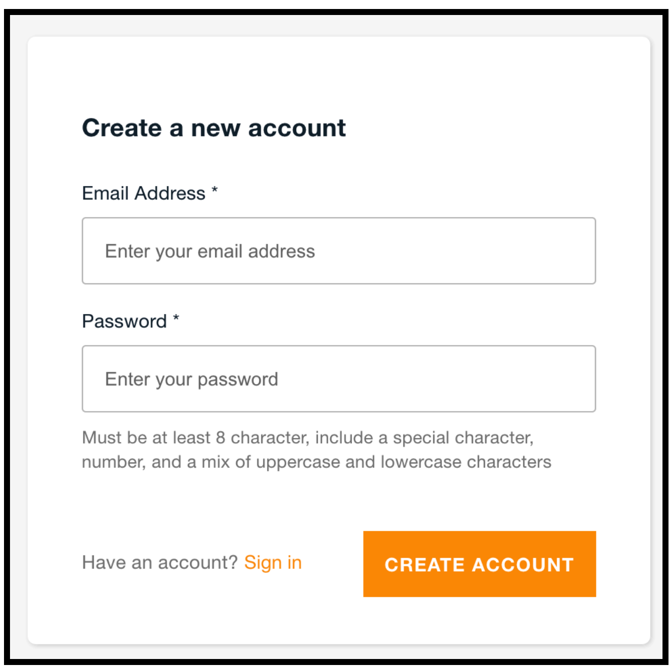
Now that you've thought about what device you might build, let's build it! However, to keep things simple (and to make sure the lab provides you with the right educational experience), instead of following your design, we'll tell you what to build. Please follow the steps below using the Illinois IoT Playground:the steps below. You will find out that parts of your design can't be implemented - the Illinois IoT Playground may not support all the sensors you need and such. That is ok! Think about what alternate components can fit your needs and use those instead.

1. Connect to the UIUC VPN (if not installed, instructions to set up the VPN can be found here). In the browser a login window will be opened at http://iot-2.cs.illinois.edu as below :

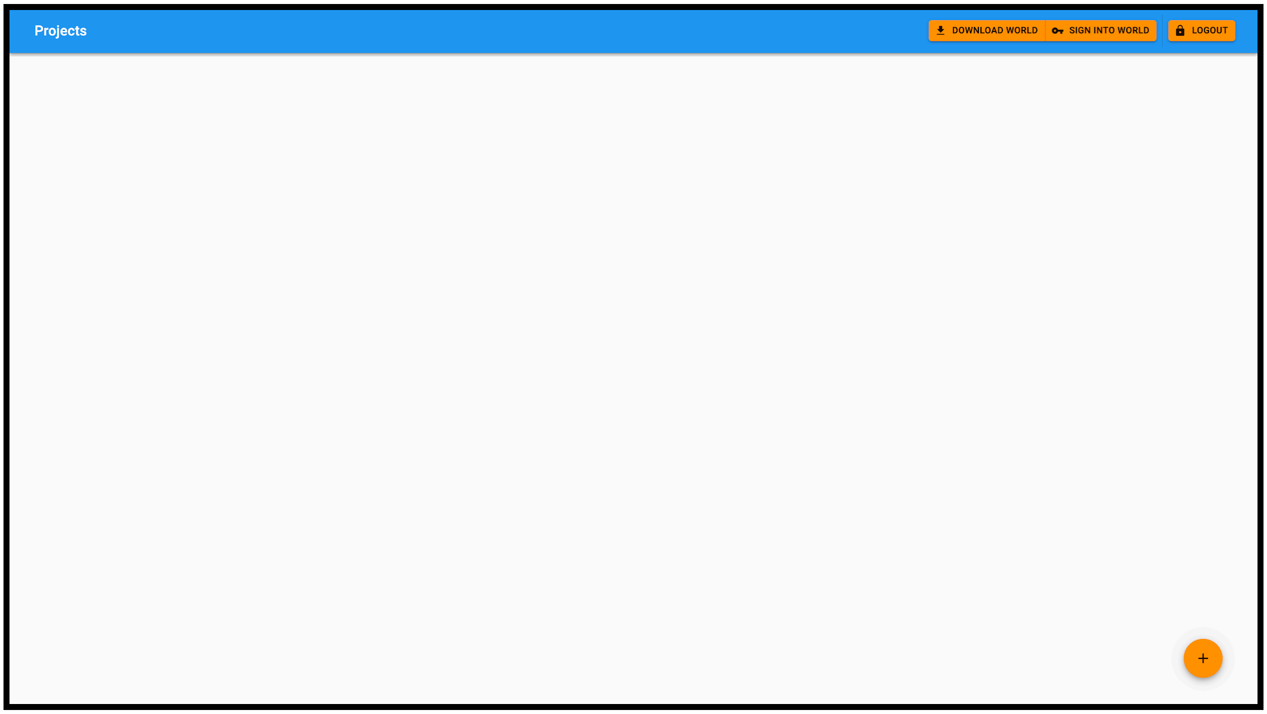


2. Click **CREATE ACCOUNT**.

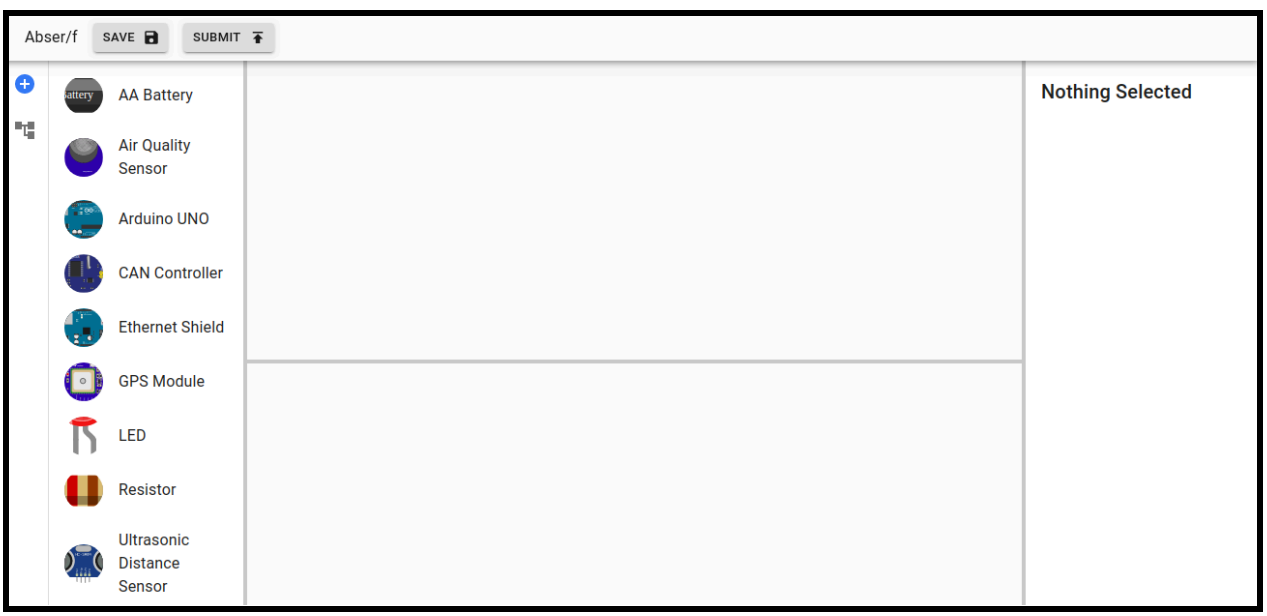
3. You'll get the following page. After filling up all the fields click the **CREATE ACCOUNT** button.



4. Now login with the newly created account. You'll be greeted with the following window.

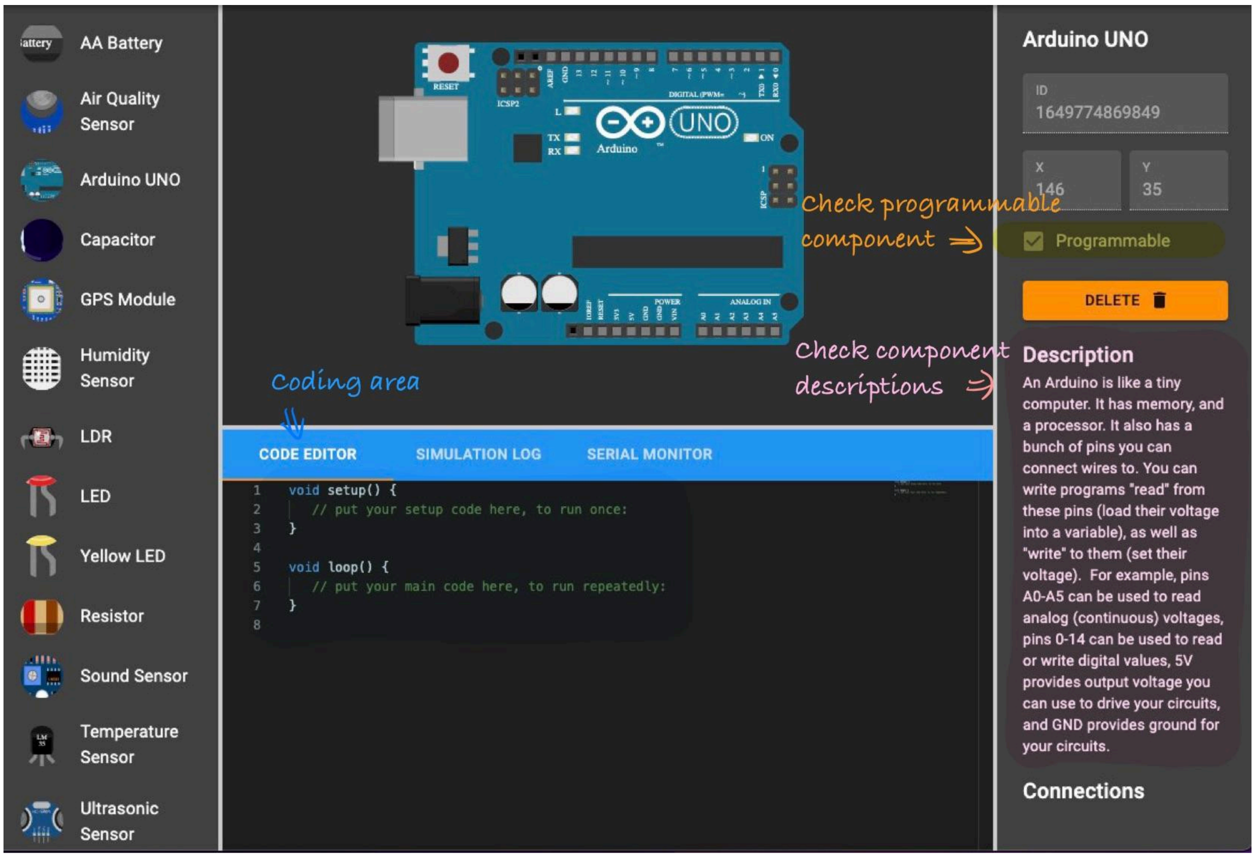


5.Click the button with plus on the bottom-right of the page to create a new project. You will be greeted with the following window:

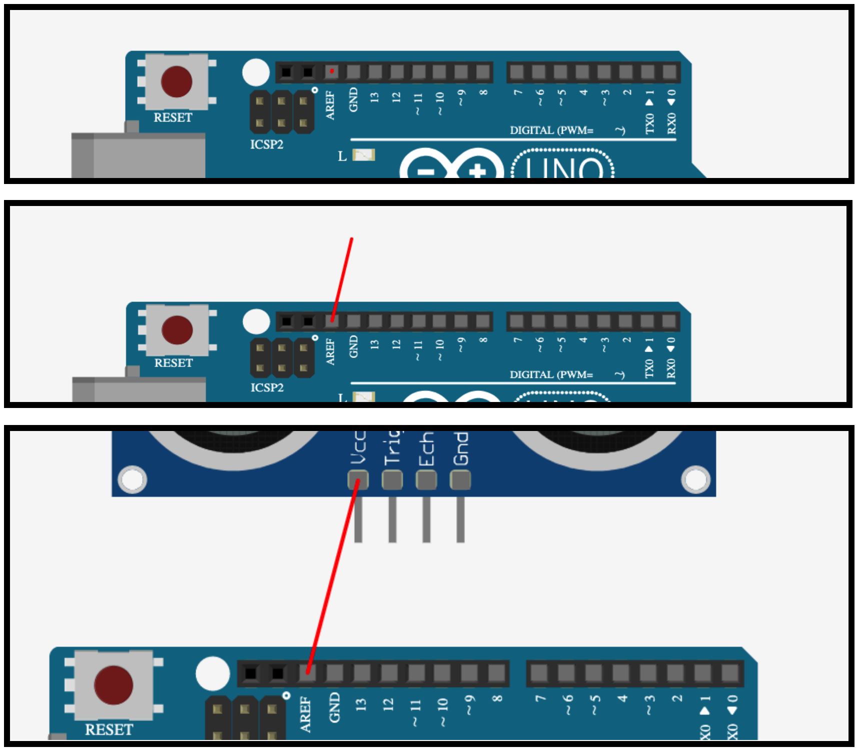


In the left pane you will see a list of components that you can use to build your device. From the left pane you can drag and drop the component to the upper middle part of the window (or click and one will be placed out for you).

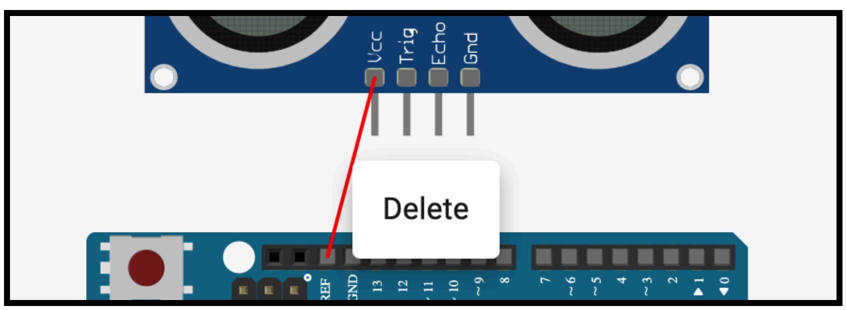
From the right pane you will be able to see which components are programmable. **Take some time to read through the component description and make sure you understand each component's property thoroughly**, this is critical for you to design and debug IoT devices later. You can write code in the lower middle part of the window to configure the component.



To add a new wire, click the pin that you want your wire to start from. A red line will show up from the place you click to the current position of your pointer. Click another pin to form a connection.

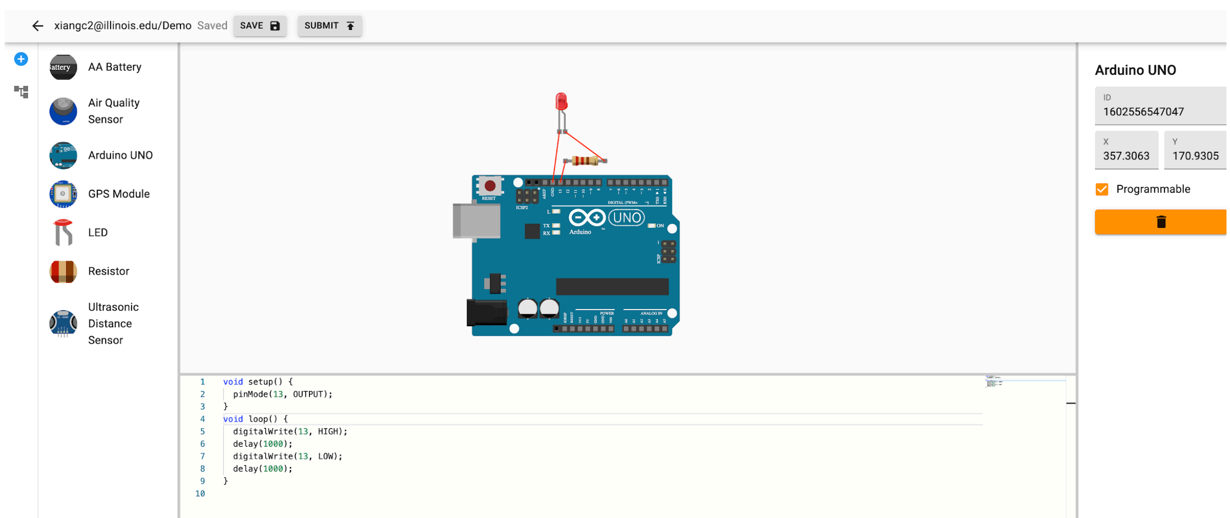


Right click the wire to delete it.

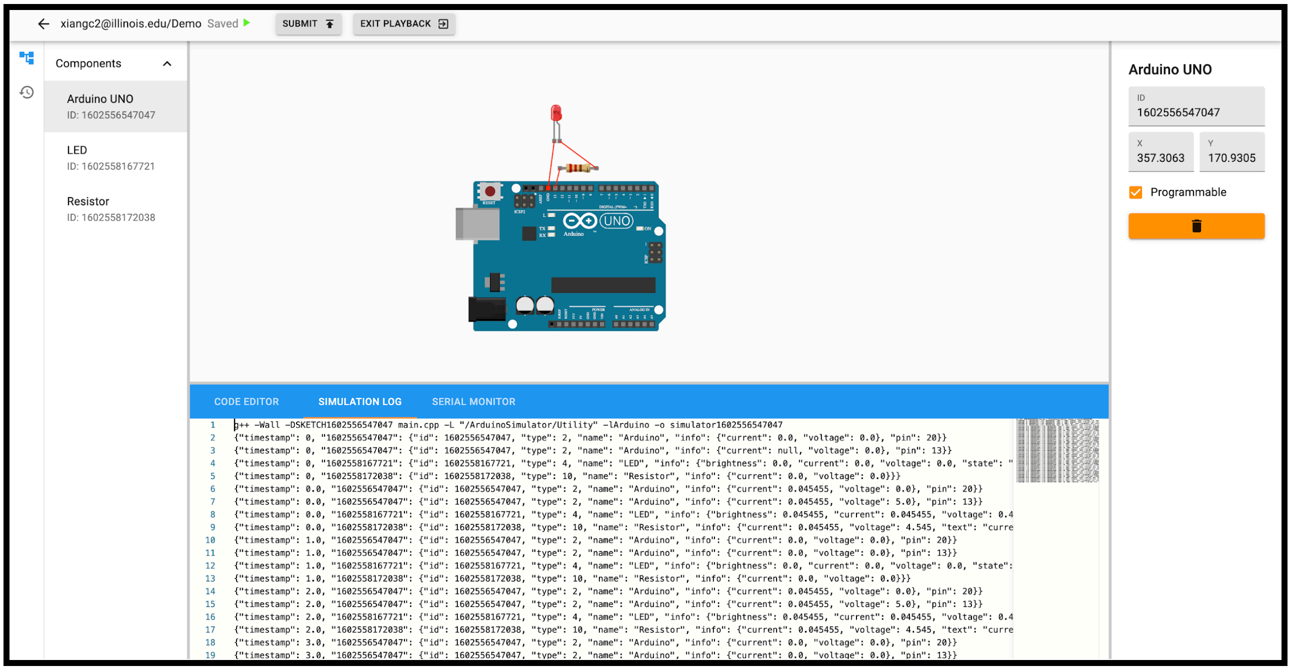


Components can be deleted by selecting the component and clicking the delete button in the right pane.

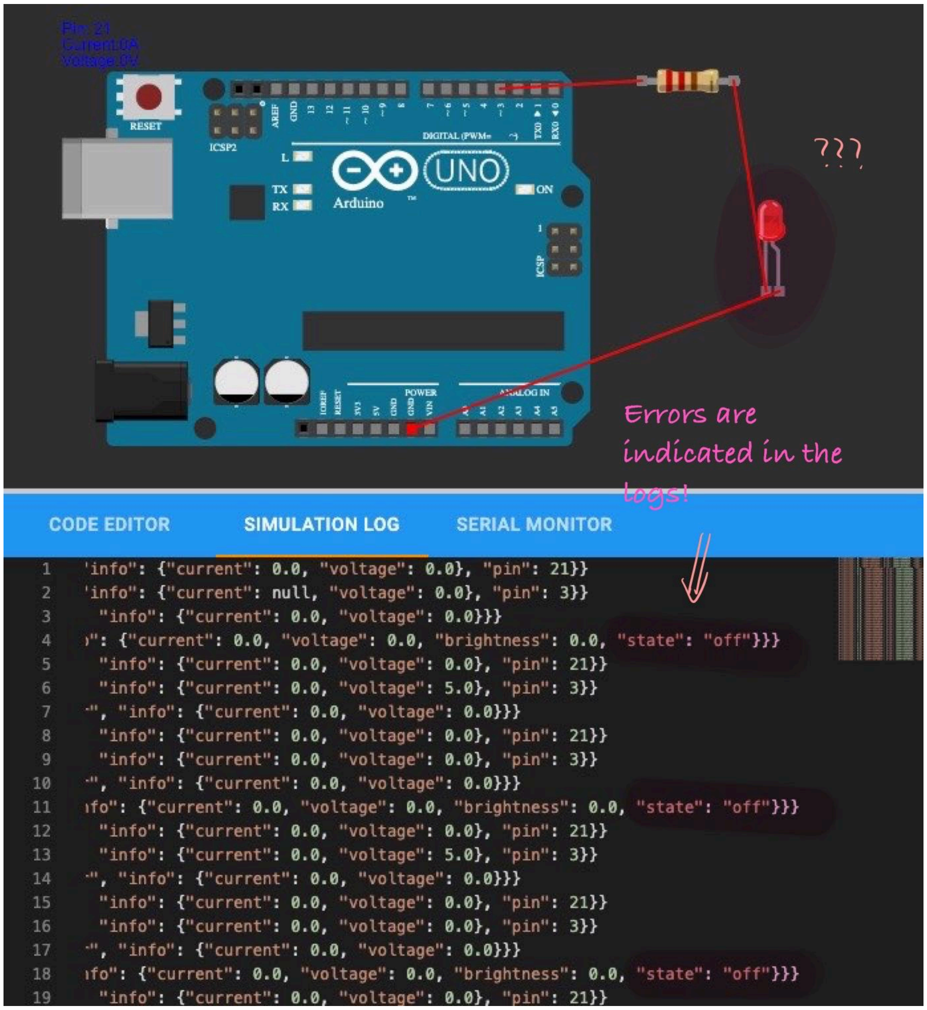
Go ahead and play with it a bit - build a simple circuit and test your circuit connection first before you start to program further functionalities with Arduino. After verifying the circuit, adding functionalities/behavior you would like your device to do in the coding area. After this, your screen will look something like:



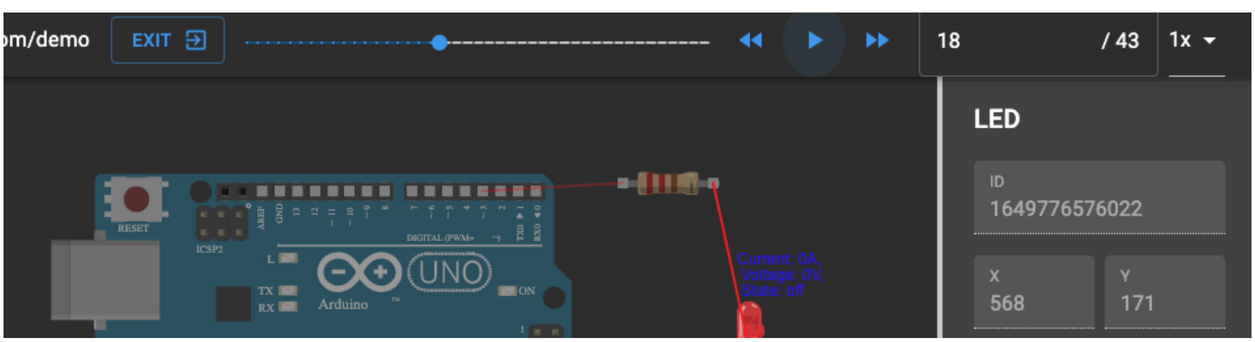
Then you can save or submit the circuit by using the buttons at the top. After you have submitted your code and circuit, you will be able to debug your IoT device with the simulation result in the simulation log panel.



Simulation log provides a lot of useful information. Not just errors that tell you what could go wrong in your circuit. Make sure you check your sensor output and state for two to three timestamps at least. Can you see what goes wrong in the example below? (Hint: check LED component description to see the right way to wire it in a circuit! Which leg is the positive side?)



The platform also provides simulation animation for you to check graphical display along with simulation results. Try playing animation, going to a certain timestamp and changing the animation speed if you want.



Step 2c: Implement your device

Next, we need to create a device to attach to the animals. To do this, log into the web frontend. Create a new project that contains a GPS, vitals sensor, and at least two other sensors of your choice and connect them back to an Arduino. The document IoT Playground Circuit Configuration Tips provides some helpful tips and tutorials for you to set your device up. You can only have one Arduino per Project in the IoT Playground app, but you can have multiple Projects each with different circuits inside. Program the Arduino to acquire readings from the sensors. Record periodic samples of animal location, activity (sleeping, standing, walking, running) (hint - to determine activity you will need to watch certain sensors for about 2 to 3 hours up to a maximum of 5 hours). Record them fast enough so that you get representative information about these activities. Also fast enough so that you can determine if animals "encounter" each other, to track their interactions with each other.

Please note that when programming, a “delay(100);” line MUST be added in your loop() function, to ensure proper circuit simulation in our backend infrastructure.

In later steps, you will need to observe output from your devices - please use console logging (Serial.println) to record information. In later steps, you will be able to observe console output from your devices as you walk around. Another thing to keep in mind when you wire up your device is **having a power source (battery)**! After being deployed in real-life, IoT devices would need a power source to support data gathering and transferring.

Step 2d: Network your devices (Extra Credit)

Your sensors will be deployed on animals walking around. If something goes wrong, it's not like you can go out there and replace them - you will lose data. Also when you want to collect data what are you going to do - it's not like you can walk around and find every single zebra, they will be all over the place.

To address this you will implement a robust replication scheme using gossip. Gossip protocols are used in environments where you face severe low power or scaling requirements. For example, they are used in Blockchain protocols to communicate values of nonces. In this environment, you will implement a gossip protocol to replicate data across zebras, so if you sample one, you will have a good chance of getting the data you need. Your protocol will replicate observations, as well as perform aggregation, by computing distributed sums and averages and other relevant statistics.

For now, Illinois IoT Playground only supports a low fidelity version of networking where Arduinos can directly communicate with each other. This means you will not be, for example, wiring an Xbee to connect with your Arduino. Please refer to this Networking section of this link to see an example of how to network between two Arduinos in our platform. :
https://docs.google.com/document/d/1nptq3pxRHCzVcM9MtQXpwx6GMAdnCV5i5sor6mv8O5g/edit?usp=sharing

Step 3: Deploy Your Device

SUGGESTIONThe virtual world should be downloaded from the frontend in Step 3a (link is on the web interface "Download World"). This link should be used only if you encounter problems with the builds in Step 3a. You will need to select the executable to download according to your operating system. We support Windows, Mac and Linux. WebGL is not supported at the moment.

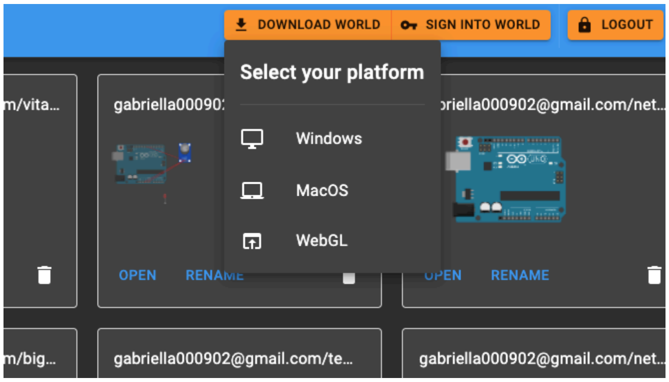
In this section, you will deploy your sensors on the zebras and all other species. To do that, you will be stepping into a virtual, electronic world, and walking amongst the animals. In this world, you will assume the role of a lead conservationist. You are able to place sensors by "clicking" them onto the zebras. Shooting a sensor instantiates a copy of your device. Do not worry, it does not hurt them, you are just using a device-attacher which is very soft and caring and the animals actually do not feel anything at all[5]. Remember that it is for their own good that you are tracking them. In real studies things are more painful than this, you may need to put the animal to sleep to get the tracking device on them, though the IoT space is advancing rapidly and ingestible and sticky trackers are on the horizon.

To enter the virtual world, you will be perform the steps below:

Step 3a: Download the virtual world

Download the executable from the top right corner according to your OS. We also have the WebGL version implemented to allow you to enter the world directly from the website.

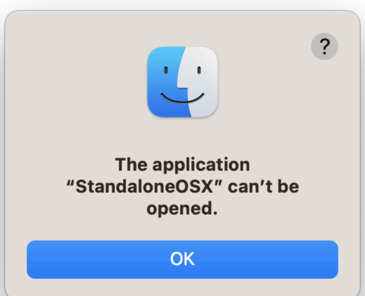
[5] Also do not worry about your own safety - the animals cannot hurt you, the elephants cannot step on you nor will the lions try to eat you. Nothing you can do can hurt the animals either, they are very safe and happy in our virtual world.



Step 3b: Start the game

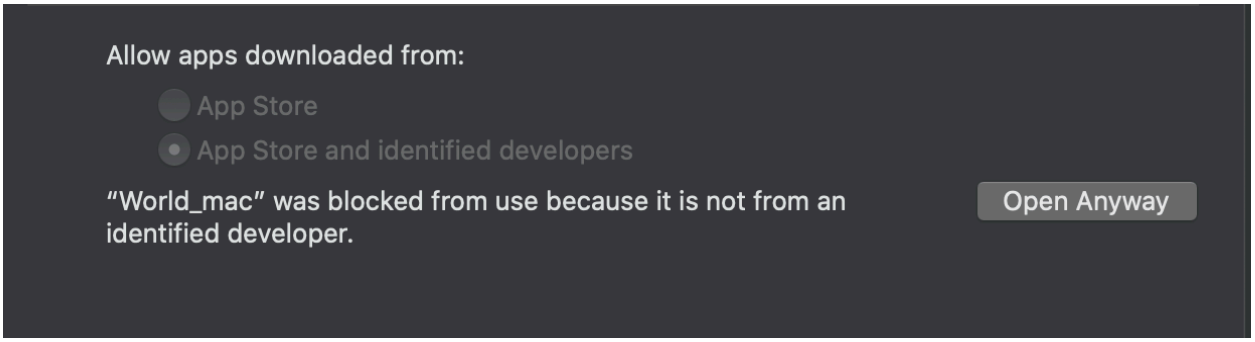
Extract files from the zip folder you downloaded from Step 3a and run the executable.

For the MacOS version, after you unzip the file, move the StandaloneOSX.app (not the folder) to a different folder of your choice. The app might not function correctly if the above step is skipped. Now try opening the app. If you see this message below:

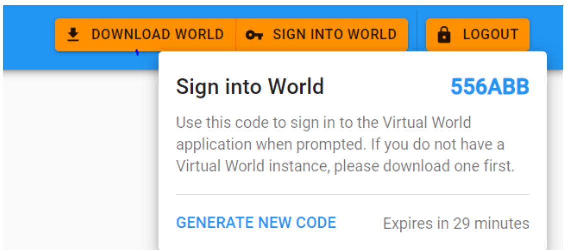


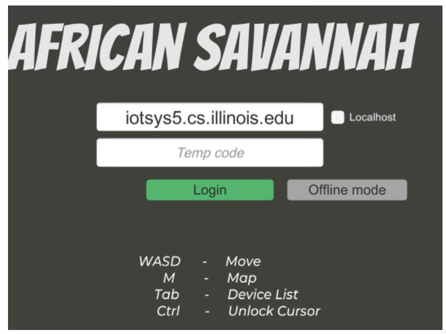
Open the terminal. Then run `chmod -R +x <project directory>`. Try to open it again.

The Gatekeeper feature on OS X may block the application because it is [not signed & notarized](). You can override this in System Preferences > Security & Privacy (shown below) or [use this approach for opening the app]().



Login with the IP address iot-2.cs.illinois.edu(which should be automatically filled in) and a Temp code which can be obtained from the web frontend by clicking “SIGN INTO WORLD”.





Then you will find yourself in the middle of the Savannah

Step 3c: Game instructions

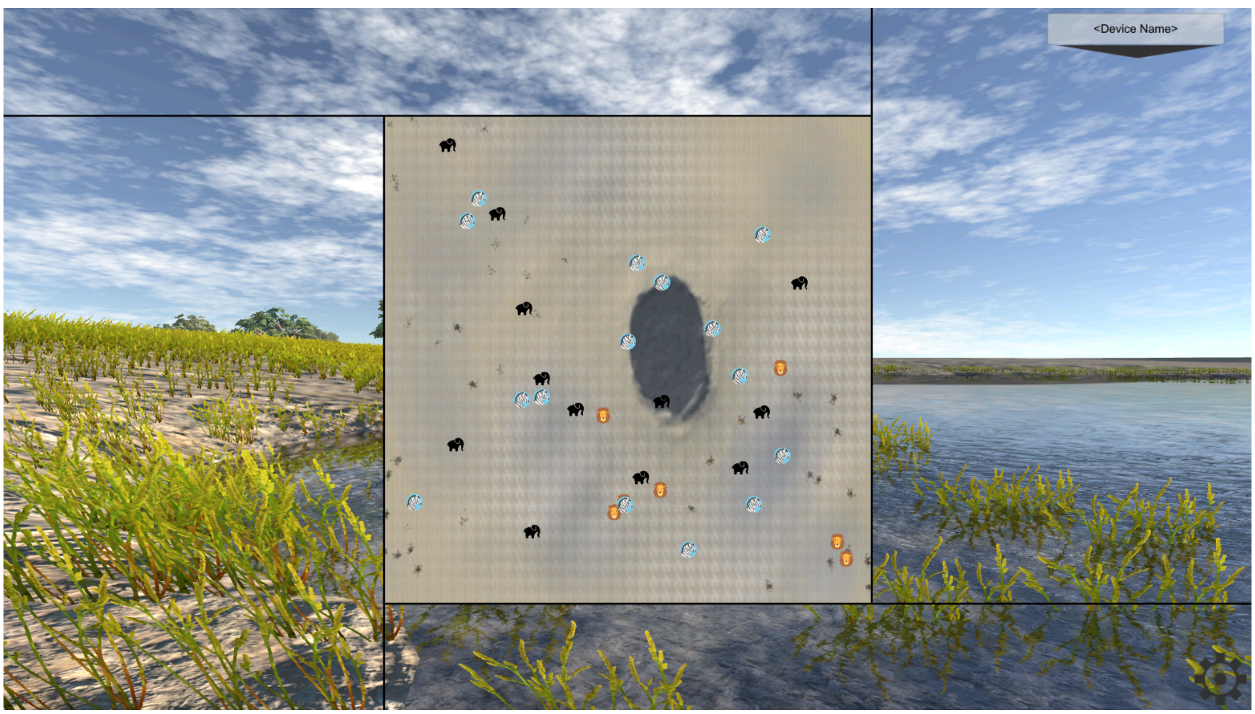
If you've played video games like minecraft before, you may already be familiar with using a keyboard to control your position and motion within a game. Our controls are not too different from games like minecraft.

When you start playing, the game will "capture" your cursor (so you can use the mouse for various operations) and allow you to adjust your camera view. To release that control, hit the CTRL key (on Mac and Windows), or the E key (on WebGL) on your keyboard. You can readjust your view by pressing the CTRL/E key again based on your OS. You will also need to release control to make a selection from the device or animal menu after opening those.

To move around the game you can use the arrow keys, or the W, A, S, and D keys on your keyboard. You can also use the mouse to rotate yourself, if you want to go in different directions.

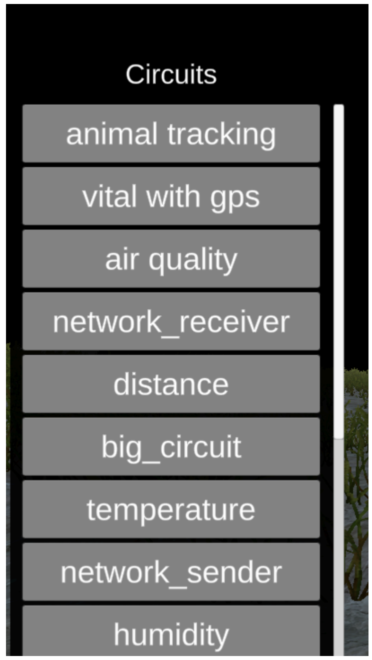
We also have a special magical map that you can use to magically teleport yourself to different places! That can be faster than just walking everywhere. To activate your map, hit the M key on your keyboard, then use the mouse to click where you want to be teleported to. Here is what the map looks like:

Map: Toggle M to open or close the map and click a place to teleport. Toggle M again to close it.



Now, comes the fun part - you can actually take the devices you created, and stick them on the animals!

We have also created a special magical console which can display all devices. To pull that up, hit the tab key, and you will see something like this:



You can then click on the name of your devices to attach them to animals with an animated cursor as a guide. Once a device is chosen, hit tab one more time to close the list and click on the animal to attach the device. The device will automatically begin running. You can see the simulation result details by getting closer and moving your cursor on to the animal.

For ease of later analysis, your console output is written to disk located in the folder "SimulationLogs" with the same root as where your executable is located. Within this folder are files to help you decode the animals and sensors as well as logs of the simulation in a readable and JSON format. Use these files to determine animal data.

Step 4: Analyze Your Data

You now have the ability to create IoT devices with sensors, place them on the animals, and use them to conduct studies.

Go ahead and try it out. Output some data and watch what the animals do by post-processing the data. What do you notice? Do you notice any interesting behaviors? (To make it easier to analyze the result, you may want to parse the simulation result into json files. Here is a python code that you can use for it.)

```
"""
CS 437 Lab 5: Wildlife Conservation Data Extraction Script
Author: Blake McBride (blakepm2@illinois.edu)

Purpose: To extract data from the READABLE simulation logs into a clean .json
format and facilitate analysis
"""

# import re for regex and json for saving the data
import json
import re

# method for parsing only the GPS data at each timestamp for each distinct animal
using regex
def parse_gps_data(file_path : str) -> dict:
    """
    Parses GPS data at each timestamp for each distinct animal from a READABLE
simulation log file and returns it as a dictionary

    This function reads a file from the
World_win/StandalongWindows64/SavannaLogs/Simulation/simulation_2023_xx_xx_xx_xx_RE
ADABLE.txt format
    and extracts the timestamp and GPS coordinates recorded for each distinct
animal; formatting the results into a dictionary which can then
    be processed into a clean .json file at the end of this script.
```

```
    Args:
        file_path (str): The path to the *READABLE* .txt file containing the GPS
data for your animals.

    Returns:
        dict: A dictionary containing all parsed GPS, timestamp, and animal data
which can be saved as a clean .json and/or loaded
               into a pandas DataFrame for futher analysis.
    """

    # setup attributes for storing data and keeping track of the current (distinct)
animal
    data = {}
    current_animal = None

    # define some regular expression patterns for grabbing the animal id, the
timestamp, and the gps coordinates
    animal_pattern = re.compile(r'^=============== (.*?):(.*?) LOG
===============$')
    timestamp_pattern = re.compile(r'-- Timestamp: ([\d.]+)')
    location_pattern = re.compile(r'"location": \[([-?\d.]+),([-?\d.]+)\]')

    # open the *READABLE* simulation log file
    with open(file_path, 'r') as file:

        for line in file:

            # check for animal type and id and create new entries if (new) distinct
animal
            animal_match = animal_pattern.match(line.strip())
            if animal_match:
                animal_type = animal_match.group(1)
                animal_id = animal_match.group(2)
                current_animal = f"{animal_type}:{animal_id}"

                # set up attributes reflecting the timestamps and gps coordinates
for the distinct animal represented as lists
                data[current_animal] = {"timestamp": [], "gps coordinates": []}
                continue

            # check for timestamp and location
            if current_animal:
                timestamp_match = timestamp_pattern.match(line.strip())
                location_match = location_pattern.match(line.strip())

                # if there is a new timestamp, we add it to the list of timestamps
for the distinct animal
```

```
                if timestamp_match:
                    current_timestamp = timestamp_match.group(1)
                    data[current_animal]["timestamp"].append(current_timestamp)

                # if there is a location match, we perform the same operation with
the gps coordinates
                elif location_match:
                    location = (location_match.group(1), location_match.group(2))
                    data[current_animal]["gps coordinates"].append(location)

    return data

# example usage

#TODO fill in your infile and outfile paths
infile_path = "your/infile/path.txt"
outfile_path = "your/outfile/path"

# parse the READABLE simulation log
parsed_data = parse_gps_data(infile_path)

# save the loaded data into a clean .json file
with open(f"{outfile_path}.json", 'w') as outfile:
    json.dump(parsed_data, outfile, indent=4)
    outfile.close()
```

Please answer the following questions. These questions may have multiple correct answers. You need to build a case based on the data. Pretend you are a scientist and think critically. This may be your first time thinking about open-ended questions like this. Do not be afraid, you can do it[6]. Just step back and think about how you would go about answering the question, then design experiments on the zebras to figure those things out.

1. Is the Zebra population healthy? Make a thoughtful case one way or another.
2. Do the Zebras have enough room to move around in?
3. Do you see any signs of poachers? If so, where are they?
4. Plot a CDF of the movement speed of Zebras. What do you observe?
5. Do Zebras make friends? Do you see pairs that tend to stay together?
6. What locations do Zebras tend to congregate at? Why do they tend to go there?
7. Are there any locations Zebras tend to avoid?

[6] This may seem silly, but actually many educational studies show that students who regularly tell themselves before exercises that they can do things achieve both faster progress and higher learning retention than students who tell themselves they cannot do things or that things are too hard.

In your report, please include multiple visualizations of the data you collect, in addition to the CDF in plot 4. One interesting diagram might be a 3D plot, with the x and y axes as location and the z axes of time, where you can graph the movement of different animal species over time.

Finally, step back and do some more experimentation with this platform. You are a scientist that cares about these animals and the health and safety of their population and their surrounding environment. What other questions do you feel should be answered? Do you want to investigate another species, like the lions or elephants? Pick a question and do an investigation of that one as well.

You just went through a simple IoT application! Be aware that in the real world, IoT applications use machine learning algorithms to analyze massive amounts of connected sensor data in the cloud. Sensors like GPS and PulseOximeter are wired up with Arduino into an IoT device, which plays a critical role as part of the IoT application.

Simulation Info

- Log files are written after closing the application
- You will only see one sensor data for one of the sensors in the in-game display
- All other sensor values will be a constant value, except the GPS value when you look at the log file

Submission and Grading

The rubric for grading that you should follow for part 2 of this lab is provided here. Each team should submit a single report with the net-ids of the team members listed in it. We list this out separately for clarity and so you can have a clear checkpoint in the middle of the lab (in previous semesters we've found it's helpful to have that).

1. **Step 1c (Extra Credit): Did the student provide a 1 paragraph summary about what they learned from the provided research paper?**
    a. If the student writes a paragraph about what they learned from the paper, they will get 5 points of extra credit
2. **Step 2a: Did the student include a diagram and a discussion of design considerations for their IoT device in their report?**
    a. If the students include a detailed sketch of their device, answer the 5 questions, and discuss the design considerations/tradeoffs in their report, they get 10 points.
    b. If the student is missing the questions or the sketch, they get 5 points.
    c. If the student is missing both, they get 0 points

3. **Step 2c: Did the student implement Arduino circuit(s) with a GPS, Vital Sensor, 2 additional sensors?** Please include screenshots of your circuit(s) in your report. In your video, be sure to include a code walkthrough. If you encounter any bugs, please document them thoroughly in your report and inform the TAs.
    a. If the student successfully utilized 4 sensors (including the 2 required ones), they get 9 points.
    b. If the student was able to partially implement networking and documented their problems, or if the student used 2-3 sensors, or was missing 1 of the required sensors, they get 4.5 points.
    c. If the student was unable to implement networking or utilized less than 2 sensors on their circuits, they get 0 points.

4. **Step 3: Did the student successfully deploy their Arduino circuits in the virtual world?** Were you able to set up the virtual world on Windows or Mac, and deploy your circuits onto animals? Please include screenshots in your report and a discussion of how many animals and which species you deployed your IoT devices on, and why. In your video, show footage of animals walking around with your devices attached. If you encounter any bugs, please document them thoroughly in your report and inform the TAs.
    a. if the student deploys their devices on a significant number of animals of each species, they get 20 points. Both the report and video must document which animals and how many animals they deployed on, and what their reasoning was.
    b. If the student deployed on a limited number of animals, or did not properly document their deployment in their report/video, then they get 10 points.
    c. If the student did not deploy their IoT devices, they get 0 points.

5. **Step 4: Did the student answer the 7 questions about the zebras plus 1 additional question?** Please be sure to include multiple visualizations from the data you collect. In your report, you should spend time thoughtfully answering each question, referencing your visualizations as needed. If you encounter any bugs, please document them thoroughly in your report and inform the TAs.
    a. If the student answered all 7 zebra questions plus 1 additional question, and included multiple, high quality data visualizations, they get 30 points.
    b. If the student did not answer 2-3 questions, or their data visualizations were of poor quality, they get 20 points.
    c. If the student did not answer 4 or more questions, or did not include any data visualizations, they get 0 points.

6. The prelab and postlab requirement will be covered in the lab instruction and the description inside google form for each section, make sure you read them carefully! (Note: you need to finish both the prelab and postlab to get the 10 points, we do not offer partial points.)

Submission/Points	Points
Prelab	5
Step 1c Extra Credit	5
Step 2a	11
Step 2c	9
Step 2d Extra Credit	5
Step 3	20
Step 4	30
Video	20
Post-lab	5
Survey	TBD
Total	**100**

Note: Please add a table to your report listing all team members and their respective contributions.

Frequently Asked Questions

1. Step 2a asks for a diagram but in step 2c you have to provide a screen shot. Isn't it redundant as you can't get any more detailed than with the simulator screen shots.
   Ans: The sketch for step 2 might not be the same as your circuit in the simulator, since the simulator doesn't have all the available sensors. This step is more on the creative side, where you get to choose what sensors would be useful and how the device should be designed to attach onto animals. The sketch doesn't need to be too detailed as long everything is labeled and we can understand what each part is.
2. (Following above question) Does the sketch need to show pin connections as well ?
   Ans: No, just a high level sketch is fine
3. Can we get a little more clarity on step 2d and the recommended approach? Is it possible to use the lo-fi version of networking available in the IoT Playground to have a single Arduino send and receive messages? I haven't been able to get it to work properly without using two devices. Surely we shouldn't have to attach two devices per Zebra?
   Ans: You need 2 devices, one as a sender and the other as a receiver. You don't need to implement a full gossip protocol, but the receiver & sender can act as both. For step 2d, I mostly just want to see messages being sent from one device to another, as long you can show it in the virtual world or the circuit simulator.
4. only step2c, step2d and step3 require video, the rest are only report, is it correct?
   Ans: Yes.
5. does step 2d only need video? any thing that we need to include in the report?
   Ans: Just video

6. Our group of 4 people did Lab 5 as a group project as mentioned in the instructions. So, for steps 2c and 3, we used 1 simulator environment(web/unity) to complete the configuration and deployment. During submission, we realized that log data google form requires individual logs and web app info. Do we need separate log data for each team member or 1 log data good enough for the team?
   Ans: 1 log is enough