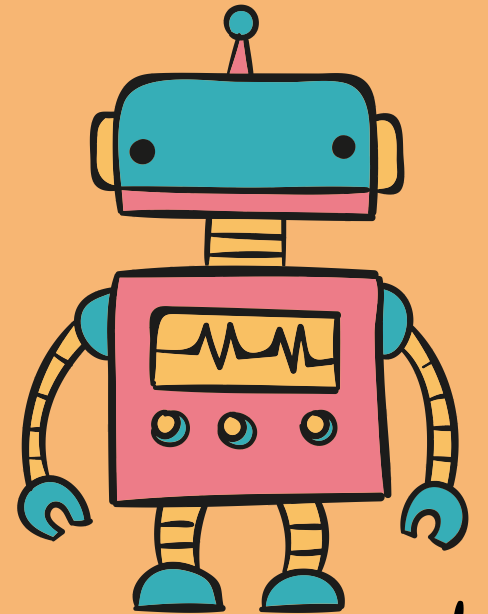
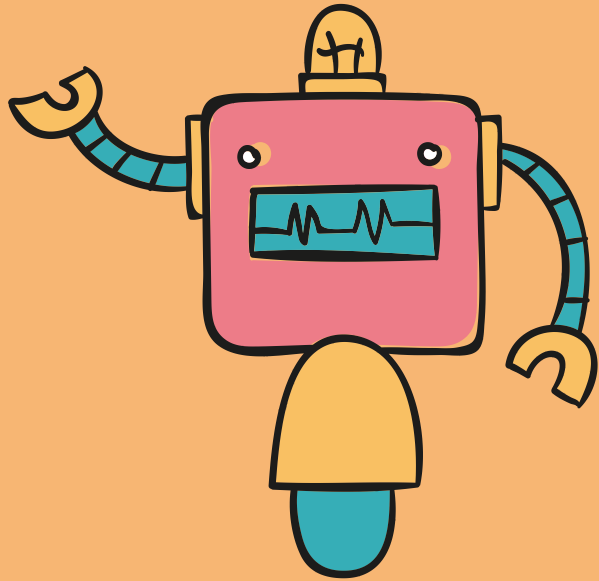


EGR 101

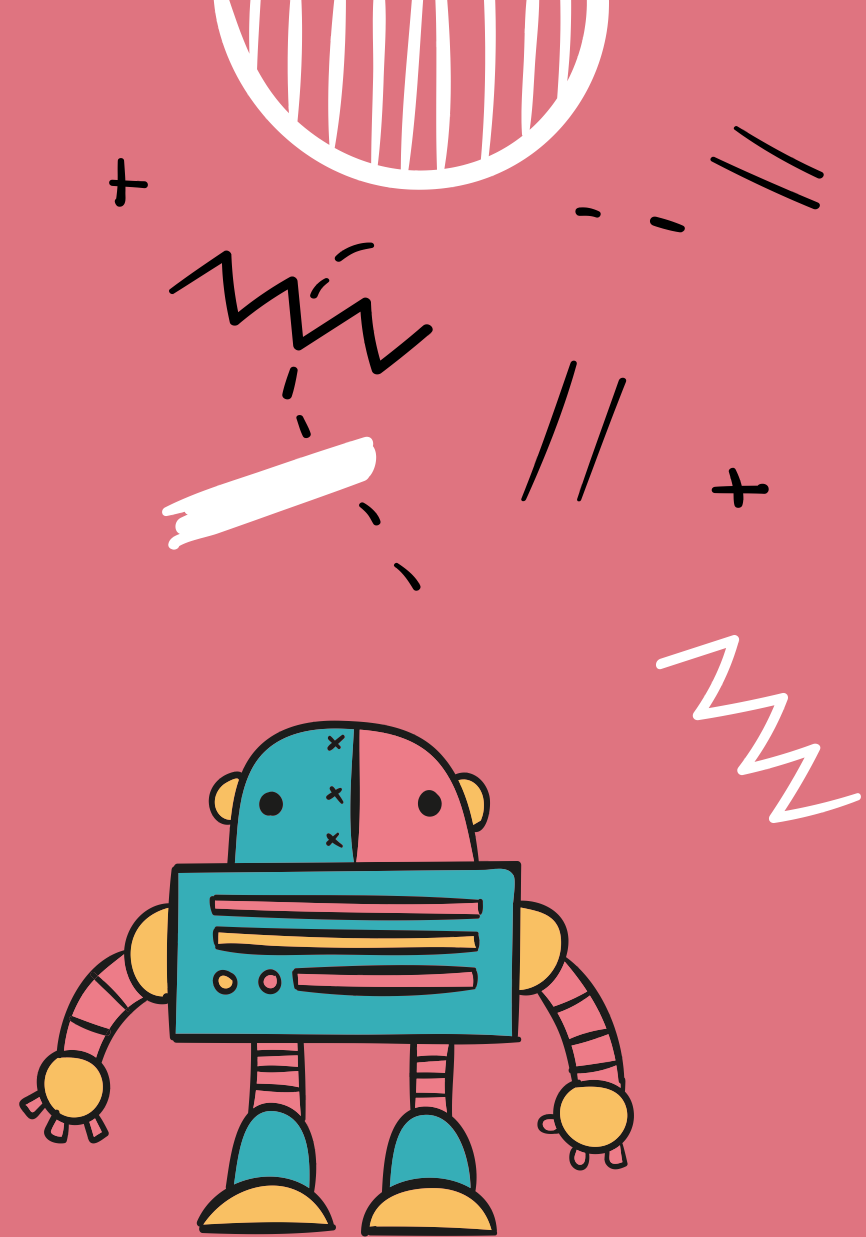
Simulation

Emily Connearney, Luke Crump, Vivian Dang,
Daniel Khalil, Keely Mashburn



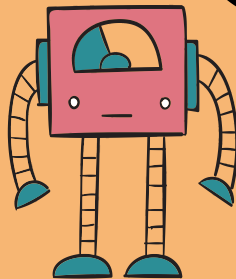
Project Description

- A downloadable software package that students can utilize to simulate Boe-Bot behaviors, thus removing the need for the physical components of the EGR 101 course.



Project Use

- Due to the pandemic, students could not do hands on coursework for EGR 101 for fear of violating COVID-19 restrictions.
- Solution was to have students purchase their own \$85 kits to do the course individually. This is obviously not comparable to what EGR 101 actually is like.
- EGR 101 Simulation attempts to emulate what being in the class is like, from putting together a bot to coding it to line-follow.





Assumptions/Dependencies

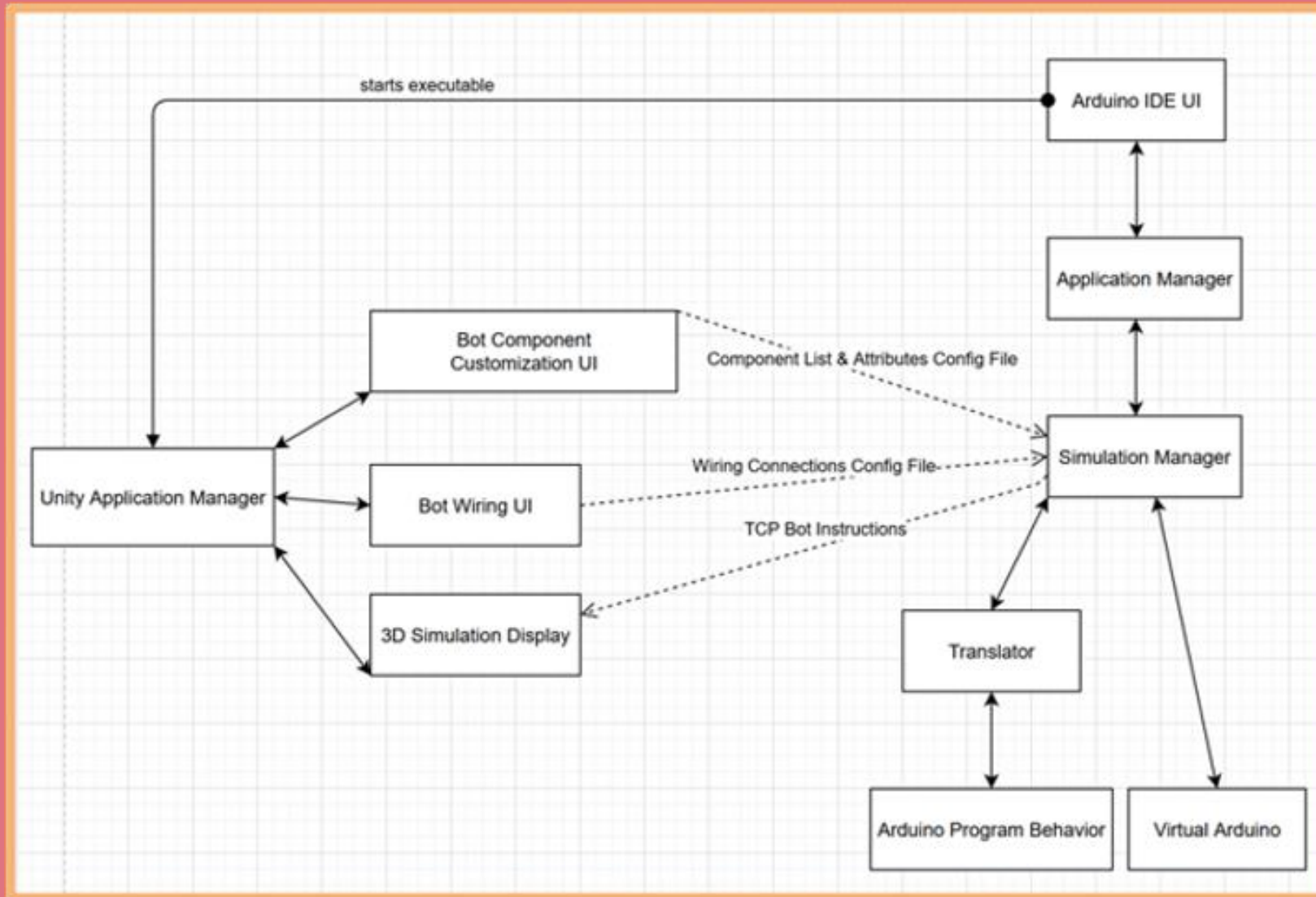
- Students are familiar with the kit and its components
- Students also understand assigned task and what to do
- Only the professor can add additional components and pieces to the kit
- The application must be able to run on different operating systems



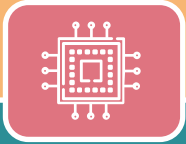
Constraints

- Project is constrained by the kit provided to the EGR 101 students
- Cannot have functionality or components that the actual robot and kit does not have

Architectural Diagram



Major Sub-Systems



Design/Wiring
Interface



Arduino IDE



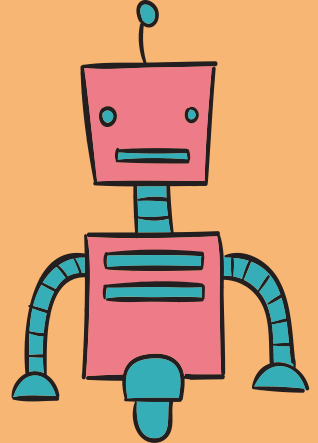
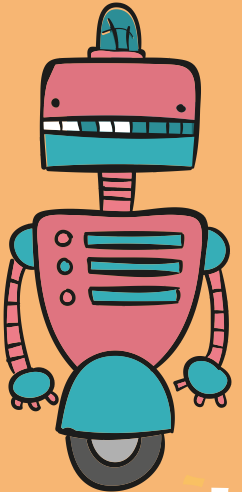
Arduino
Emulation



Communication



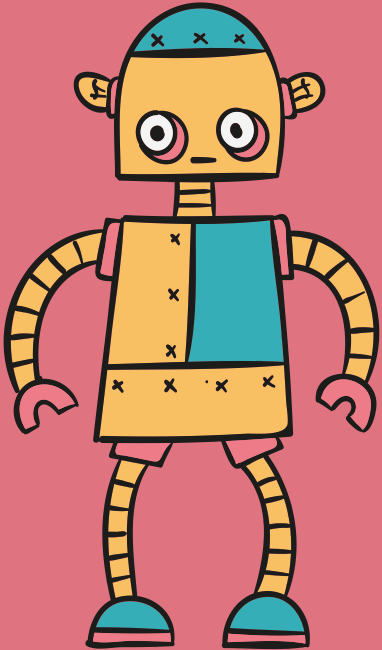
3D Simulation



+

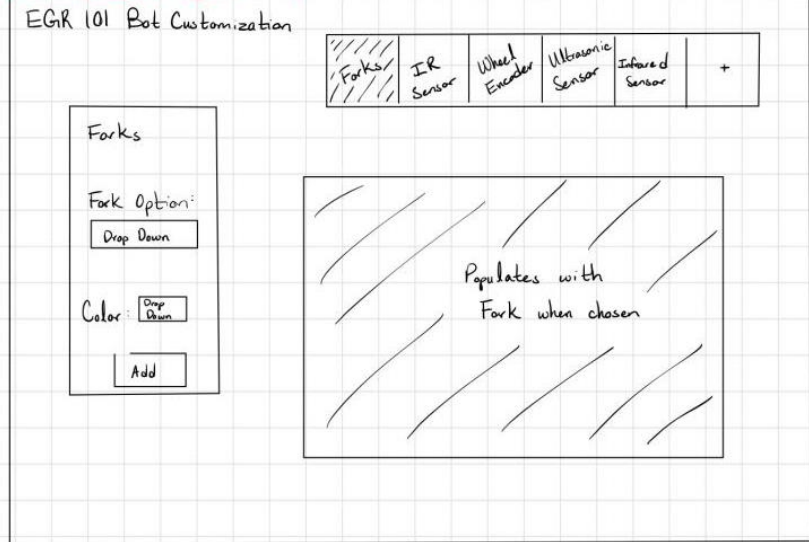
Design/Wiring Interface

- Design GUI driven to emulate 3-dimensional hardware components
- Students can customize their bot (color, sensors, frame, etc.)
- Students can digitally wire components to a virtual Arduino or breadboard
- Using Unity + C# to design GUI base

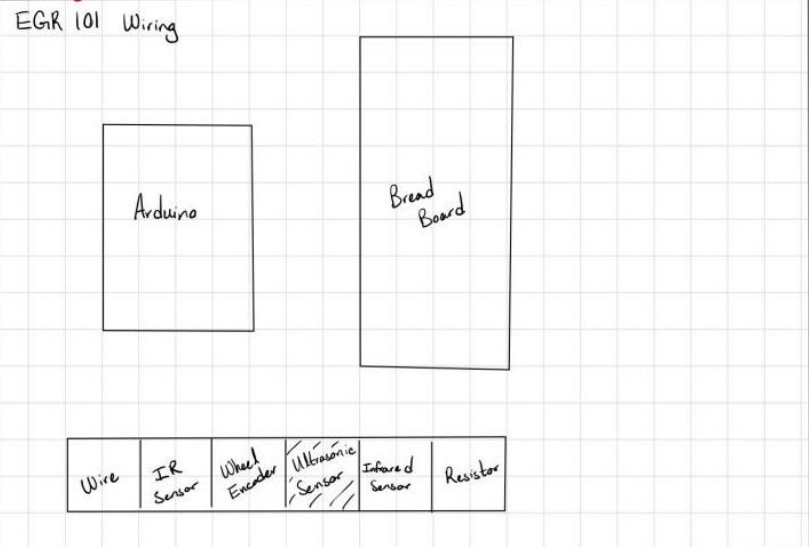


+

Bot Customization Page (Forks)

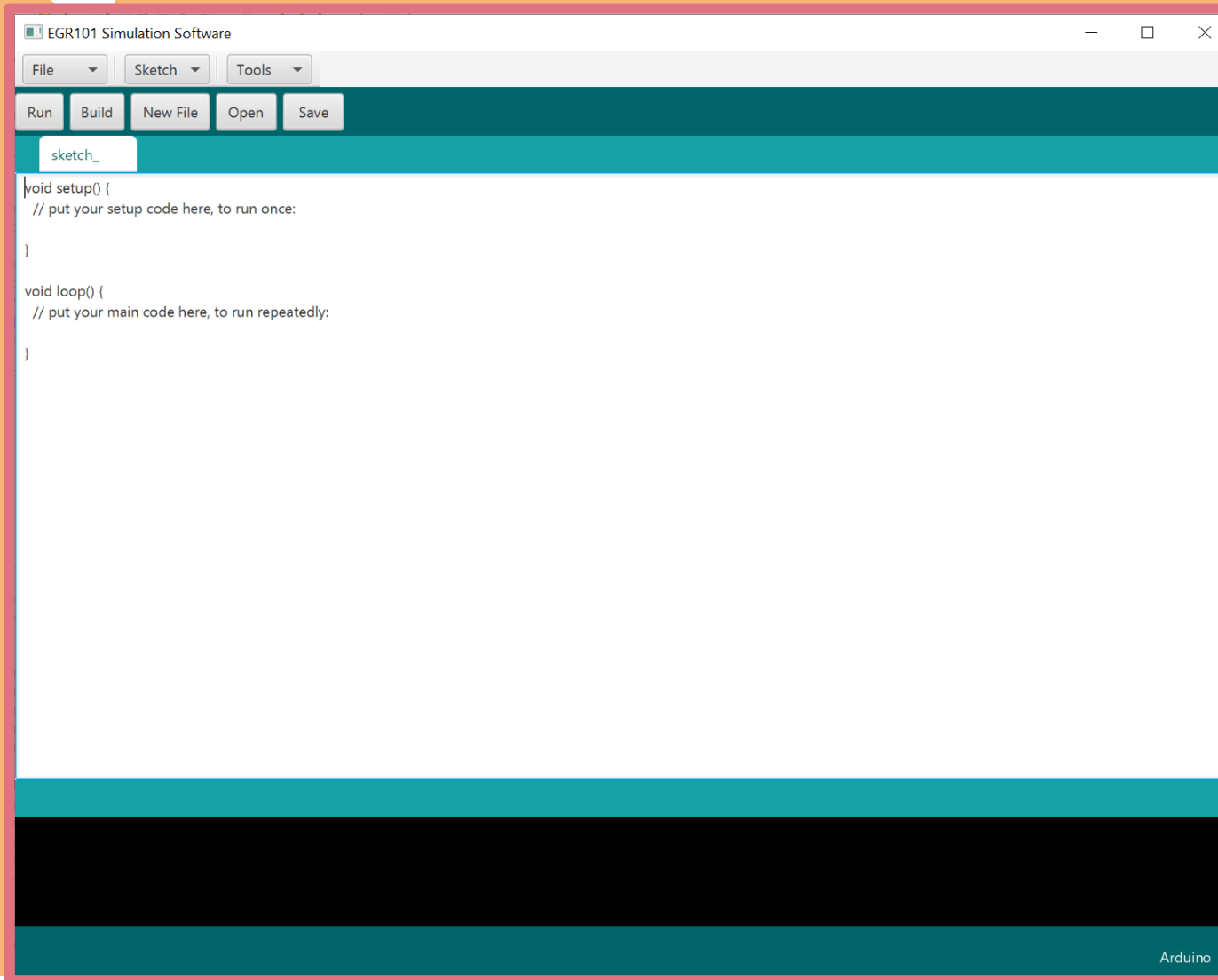


Wiring

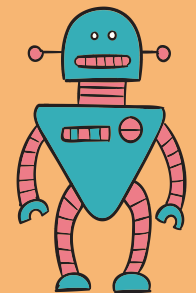


Arduino IDE

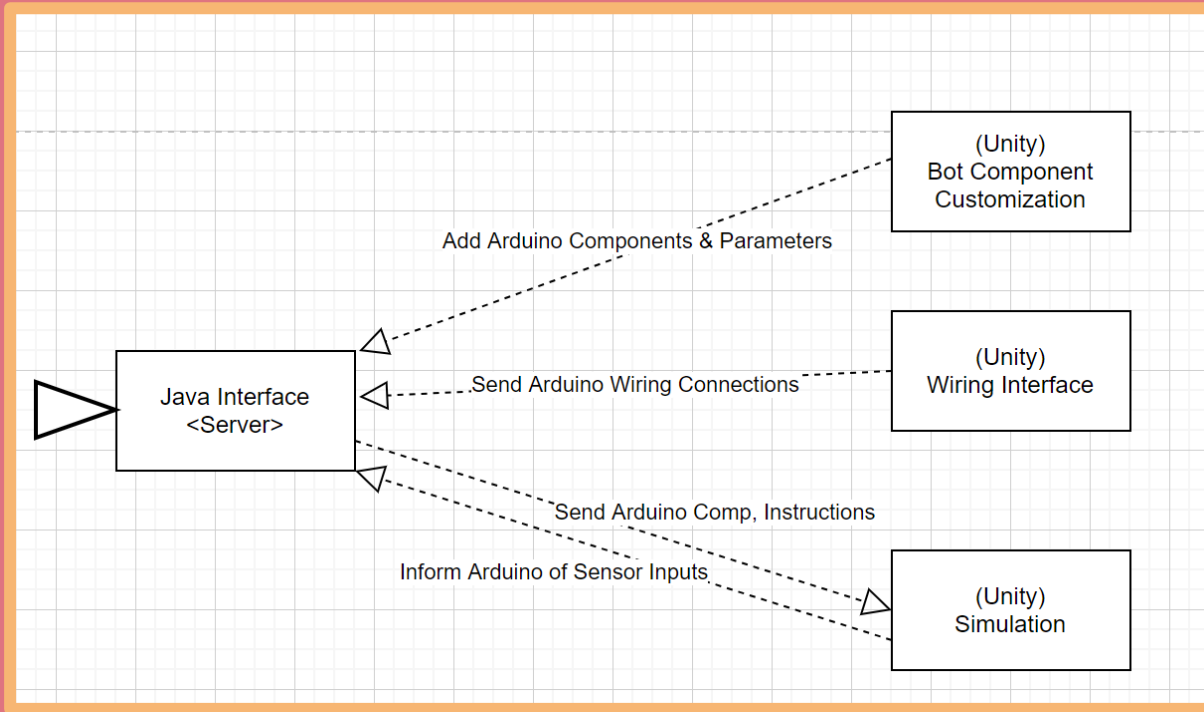
+



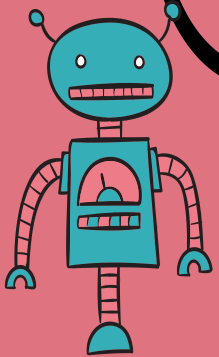
- Mimics all functionality of the Arduino interface
- Arduino code can be entered and will be translated into Java to be compiled
- This will connect to the wiring and robot components



Communication

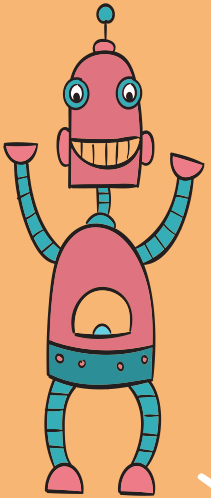


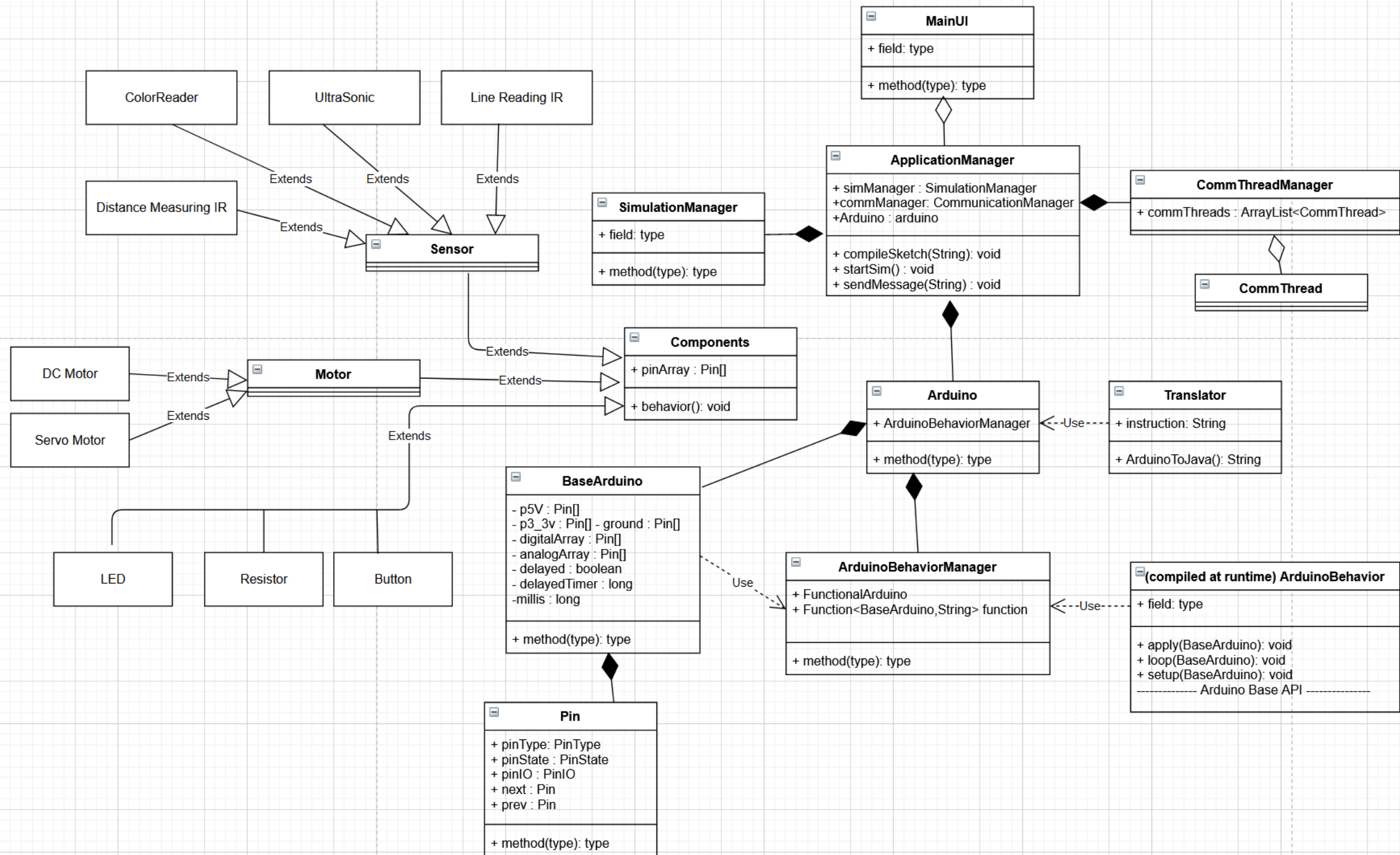
- Java Application will act as a server
- Simulated Arduino will send component info to 3D Unity simulation
- 3D Unity Wiring Interface will send wiring info to the simulated Arduino
- Bot Customization will send component addition and design to simulated Arduino



Arduino Emulation

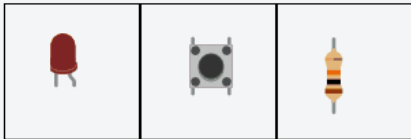
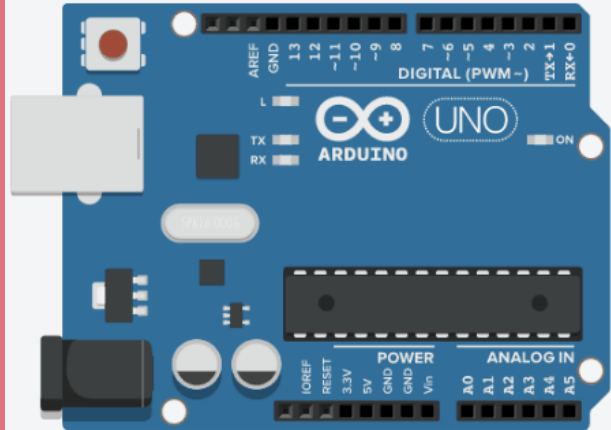
- ❖ Track Pin Connections
- ❖ Keep track of Component Behavior
 - Each component tracks:
 - Potential
 - Voltage
 - Resistance
- ❖ Use Arduino Code Translation to perform instructions
 - Delays (Simulate time in Arduino)
 - Analog/Digital Pin Writes





+ Demo 1: Simple LED Blink Test

Wiring GUI



help.jpg

EGR101 Simulation Software

File Sketch Tools

Run Build New File Open Save

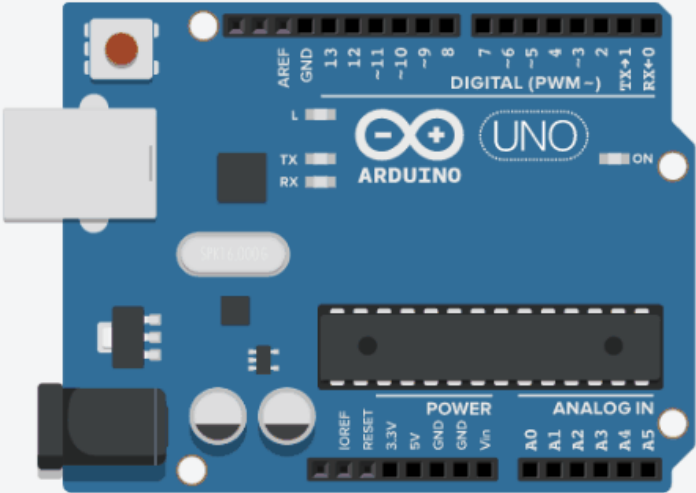
sketch_




```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(1, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(1, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(1, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);           // wait for a second  
}
```

Arduino

Demo 2: Multiple Connections

Wiring GUI





me.jpg

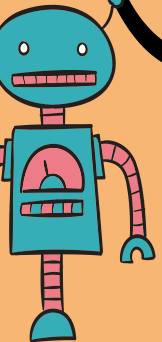
EGR101 Simulation Software

File Sketch Tools

Run Build New File Open Save

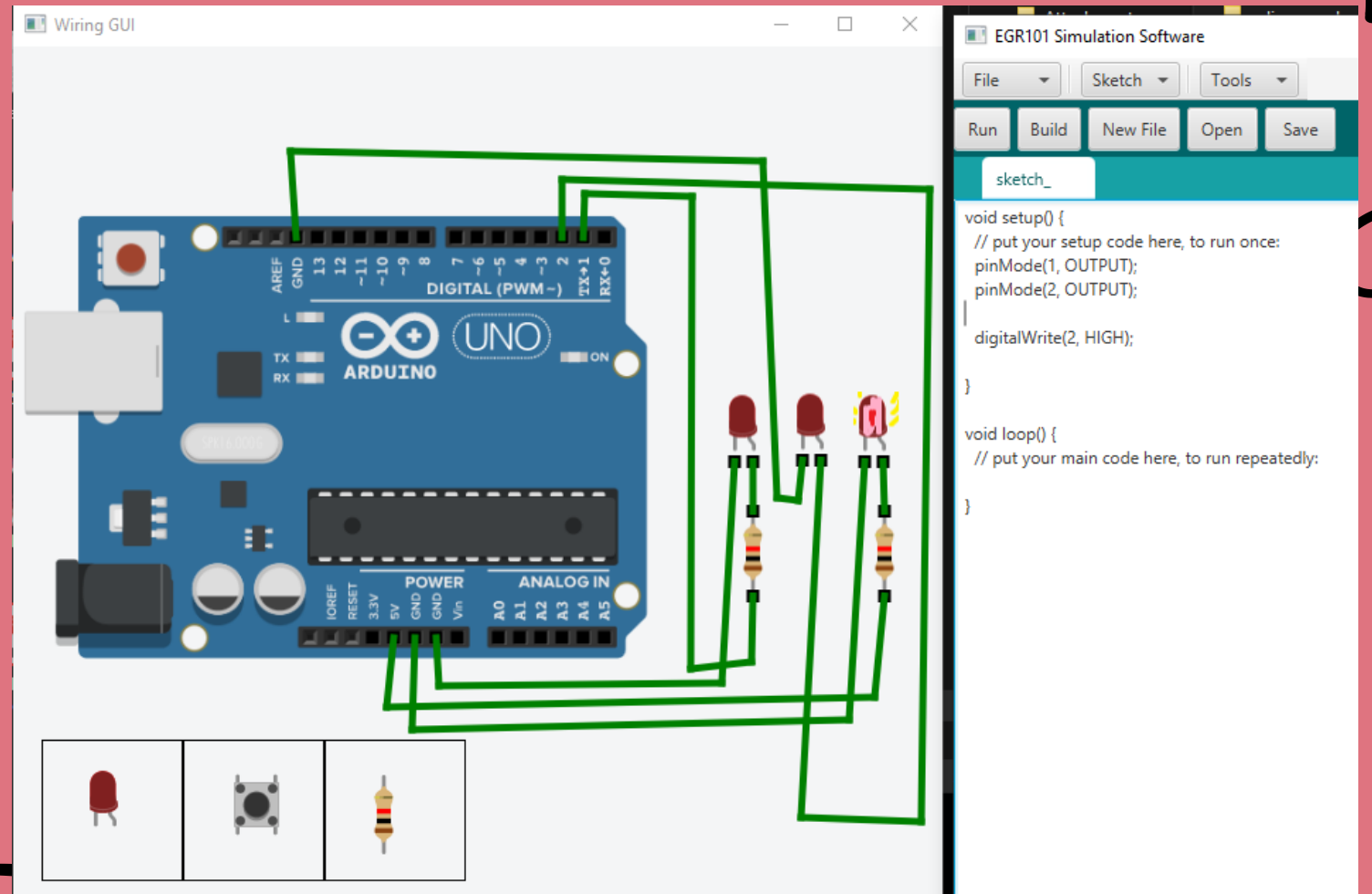
sketch_

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(1, OUTPUT);  
  pinMode(2, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(1, HIGH); // turn the LED on (HIGH is the voltage level)  
  digitalWrite(2, HIGH);  
  delay(1000);           // wait for a second  
  digitalWrite(1, LOW);  // turn the LED off by making the voltage LOW  
  digitalWrite(2, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);           // wait for a second  
}
```

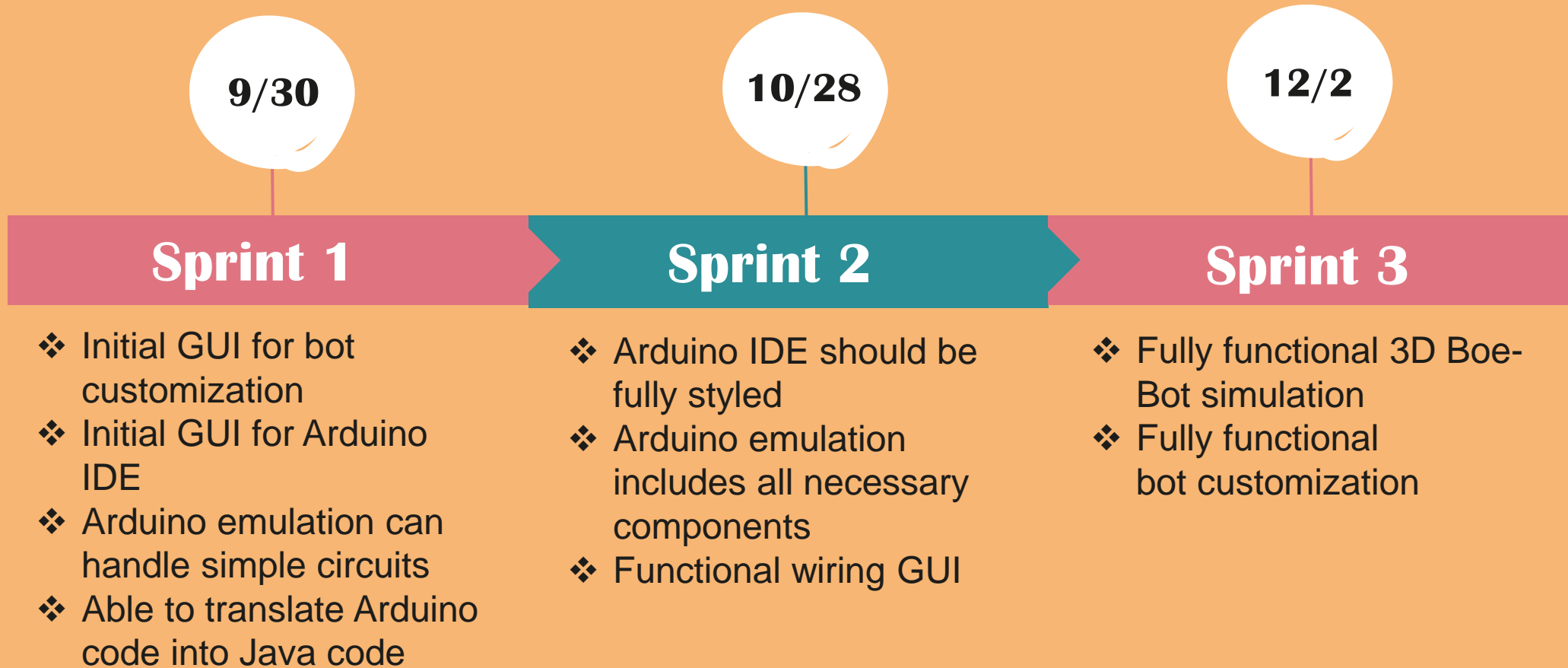


Fitting to Real Constraints

- LEDs won't light up if raw 5V is not placed with a resistor
- LEDs won't light up if there's no digital Write (on a digital Pin)
- LEDs won't light up if there's no ground

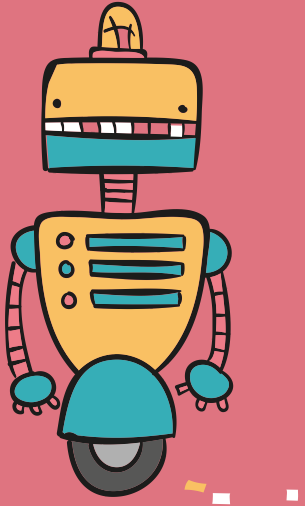


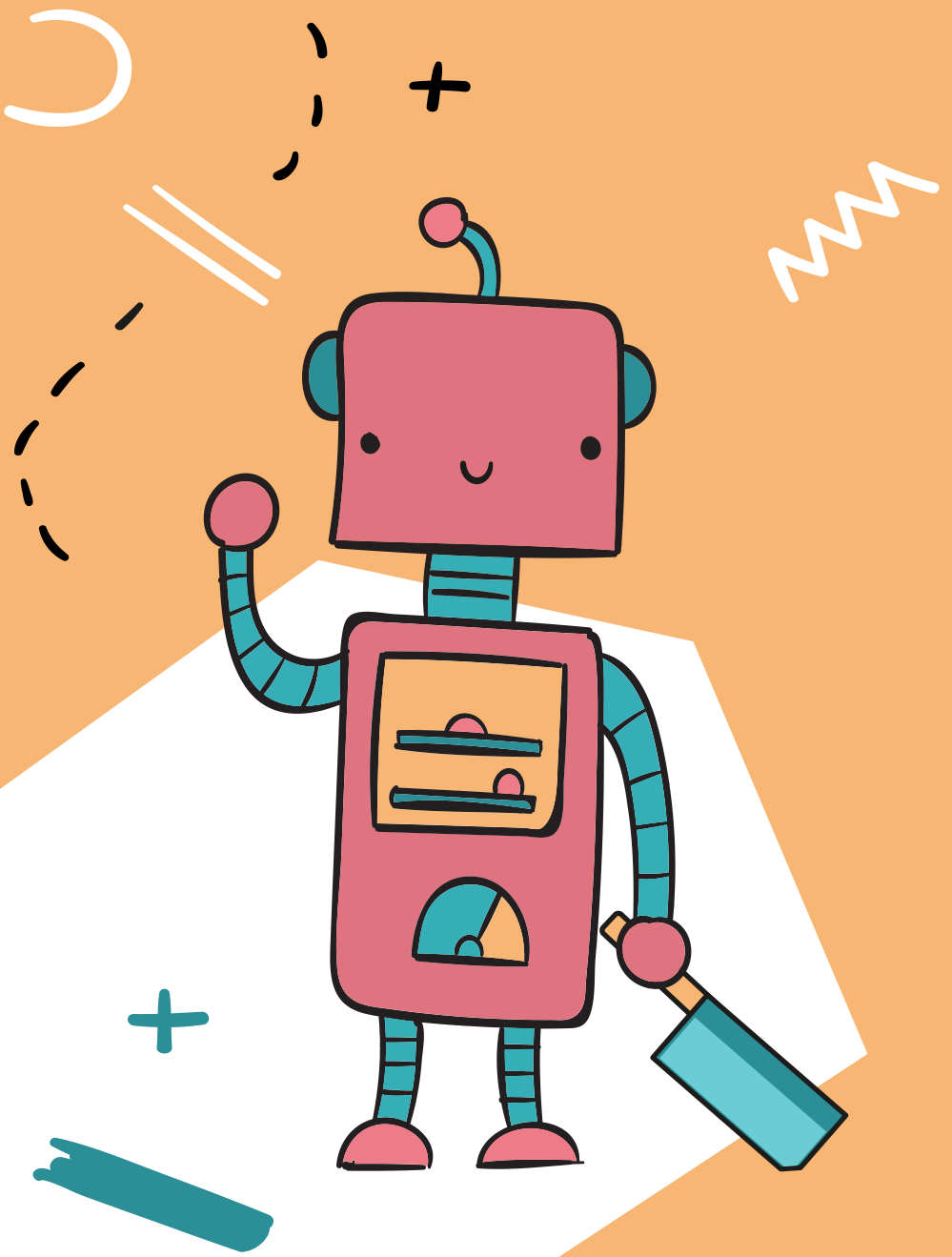
Project Timeline



Lessons Learned

- Need a form of Java/C# styling standard
- Should start defining method and class signature documentation through JavaDoc
- Properly utilizing tools like ZenHub is greatly beneficial
- Need to improve estimating time to reach project goals
- More structure and direction with team meetings





**Thank you for watching
our presentation!**

Questions?